servicenow
ADMINISTRATION

{BY}

SRINIVAS SUNKARA

About Author

Srinivas Sunkara is the **Service now Architect & Solution Provider** and founder of **Dettifoss IT Pvt.Ltd.** The education background includes in Master of Computer Application **(MCA)** from Acharya Nagarjuna University, Guntur, India. Total **10 years** of IT expertise and career was begin as Lotus Notes Developer. Previously worked on various technical platforms like... Share Point, Oracle Business Intelligence, Robotic Process Automation and past four years into Service now in software development. Training has been my passion and almost given training for **5000+ students.**

I would like to thanks my spouse **Satyavani** and son **Gautham** given great support to end this first edition of **Service Now LIVE** book and allowing to work at late night and I really appreciate their patience, way of understand, love, support, helping

I enjoyed meeting new people and hearing new perspectives. Reach out if want to talk to me about emerging technology called service now. Creating magic's with our platform. I focus on making new high-quality professional to serve our productivity

I have spent last several years utilizing my professional background as training experts and my service deliver to entire world. Currently lives in **Hyderabad,** India**,** working with **Dettifoss IT**

About Reviewers

Mr. Siva Prasad Kethineni is working as **SME** on **DevOps, Azure DevOps and AWS**. He has more than 9 years' experience in IT. He is expert in automation process and code deployment. Since 9 years providing IT services for various clients and also be a part of corporate training with me. He helped me lot for **review and edit** my content

Special thanks to **Mr. Siva** without believing in me nurturing my mind, attitude, confidence. None of this would have been possible for **LIVE book**

Mr. Saidarao Battineni is working as **VLSI** Physical Designer and Planning Static, Timing analysis, Hardware Designing, ASIC Design, ASIC Synthesis. since 10 years providing service to different clients. He got best service provider awards more times

Special thanks to **Mr. Saidarao** you helped me a lot and motivation and mind set and positive attitude.

Mr. Raghavendra is working as senior consultant and solution provider for **Share Point & Mule ESB** and working with operation teams to providing services to difference customers. Having 11 years of experience into IT and 5 years on corporate training with me.

Thanks to Mr. Raghavendra for amazing help

Content Editors & Digital Marketing

Special Thanks to

Mr. Sai Kumar Pothuguta

Mr. Raghu Ram Allamsetty

Why I am writing this book

I have been working since four years on service now platform into various applications and modules and deep look into the platform.
These all days I faced lots of struggles but did not find any solutions. There is no step by step guide available in market. Finally, I decided to write step by step instruction guide that will help to people who already working on platform also for new learners

My **Service Now LIVE** book will provide step by step instructions in topic wise and very crystal clear, New learners and student are felling happy with my book

We have worked on this practice **Service Now LIVE book**, each topic is categorized into simplified very practical steps that we use with many customers and students. Our approach has improved practice make it easy

In this book we will learn how to get hands on experience on each applications and modules. It was mainly designed for everyone who wants to become a master in service now with practical exercises

This book for whom

Who wants to become an expert in service now platform like Admins, Developers, implementators and architect's ant they will expect each topic is step by step guide lines and very clear with screen shots

Who begin recently to learn service now application and modules and who already working in organization, still anybody looking for step by step hand guide. **This book is help them 100%**

ServiceNow Live Book
First Edition

Managing Editor: **Srinivas Sunkara**

Editorial Support:

Mr. Siva Prasad Kethineni
Mr. Saidarao Battineni
Mr. Raghavendra

Cover Design & Illustrations: Mr. Sai Babu Pothuguta, Mr. Raghu Ram Allamsetty

First Edition: 2019 Aug, Printed in India

Published by
Dettifoss IT & Service Now Academy

Office Address

Flat no: 302, 3rd Floor, Namdev Block-A,
Prime Hospital Lane, Ameerpet, Hyderabad, Telangana.

Website: www.dettifossit.com

India contact: +91-9949429009

Table of Content

Chapter – 1

Service Now Introduction

What is Service Now?

1. Service Now is a cloud based ITSM platform that supports **IT service management (ITSM)** and automates common business process.
2. This software as a service (SaaS) platform contains a number of modular applications that can vary by instance and user
3. **ITSM Suit**
4. Ticketing Tool
5. **SAAS** (Software as a service)
6. **PAAS** (Platform as a service)
7. Cloud based service
8. Used by **Service Desk**
9. Providing demo environment for Practice purpose
10. Do customization with **Java Script**

Service Now Application Ecosystem

Dettifoss IT

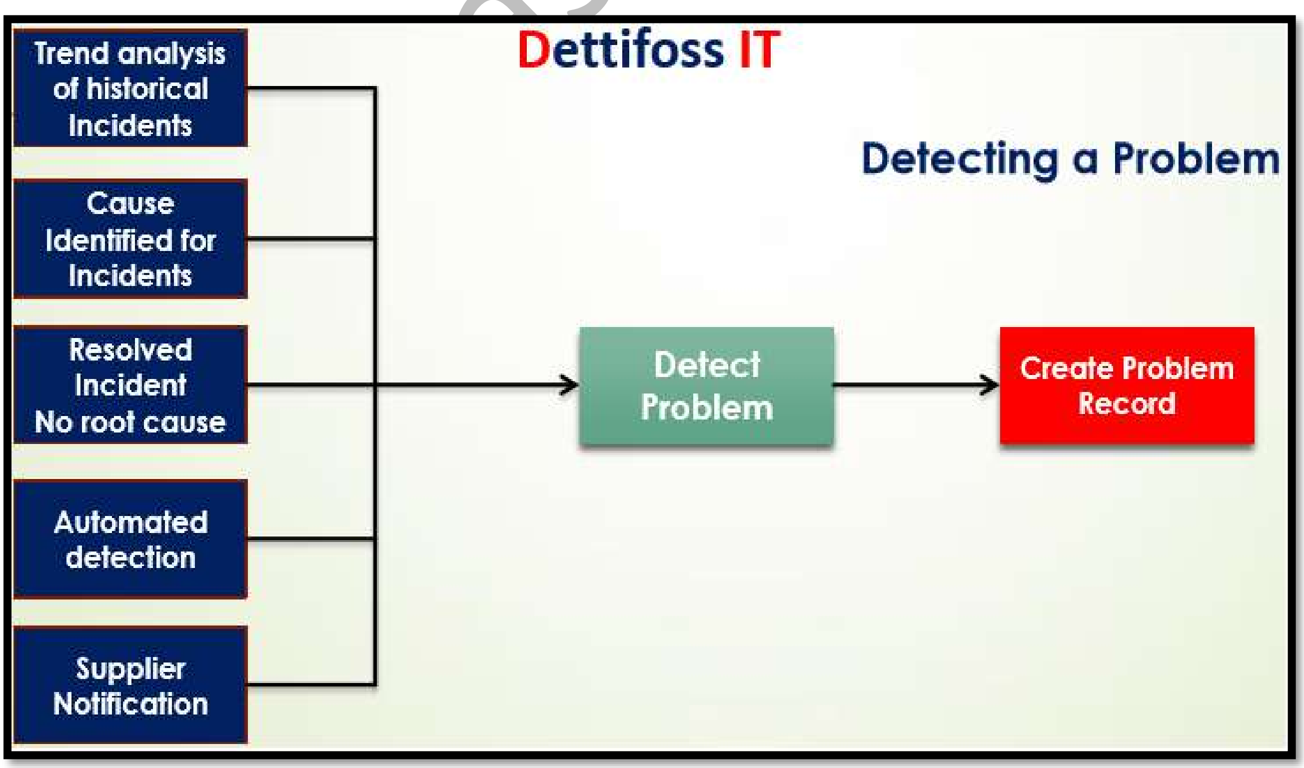
Incident	Problem	Change	CMDB	Discovery
Service Catalog	Knowledge	Asset	Product Catalog	SDLC
Facilities	Time Cards	Content Management	HR	Analytics
Governance	Vendor	Project	Contract	Orchestration
Cost	Demand	Resource	Inventory	Release

Incident Management

- ✓ Incident management is typically closely aligned with the **Service Desk**, which is the single point of contact for all users communicating with IT.
- ✓ Incidents interrupt normal service, **such as when a user's computer breaks,**
- ✓ when the **VPN won't connect,** or when the **printer jams.** These are unplanned events that require help from the service provider to restore normal function.

Problem Management

- ✓ A problem is defined by ITIL as the cause of **One or More Incidents.**
- ✓ Some incidents, such as a malfunctioning mouse at a user's workstation, are not indicative of a problem. Other incidents, such as repeated network outages, create a problem investigation due to their frequency



Change Management

- ✓ Change management (sometimes abbreviated as CM) is a collective term for all approaches to prepare and support **individuals, teams, and organizations** in making organizational change.
- ✓ It includes methods that redirect or redefine the use of **resources, business process budget allocations,** or other modes of operation that significantly change a company or organization

Release Management

- ✓ Release management is the process of **managing, planning, scheduling** and controlling a software build through different stages and environments
- ✓ including **testing and deploying** software releases.

Service Now & Improving IT Service management

- ✓ Service-Now is dedicated to improving the industry's opinion of the value software delivers.
- ✓ Service-Now develops, implements and supports a suite of on-demand IT service management application offered via the internet as software as a service (SaaS).
- ✓ These applications are licensed on a simple and economical subscription basis
- ✓ Service-Now is redefining the IT management market, providing a new opportunity for organizations to fix their age-old problems
- ✓ Effective IT service management is all about implementing process with as much automation as possible,
- ✓ Remaining flexible to your business requirements and responding to reporting and workflow needs

How Service-Now is different from other competitors

- ✓ Automatic upgrades
- ✓ Pure internet Platform
- ✓ Redundant and resilient data centers in Australia with application data encryption
- ✓ Native mobile interface
- ✓ **Simplified, flexible** subscription to all applications
- ✓ Google-like **global search** of IT
- ✓ Custom self-service portal modeled after your corporate Website
- ✓ Easily configurable applications offered as SaaS
- ✓ Drag-and-Drop graphical workflow
- ✓ **Delegated administration**
- ✓ Un presented growth, financial stability, customer satisfaction

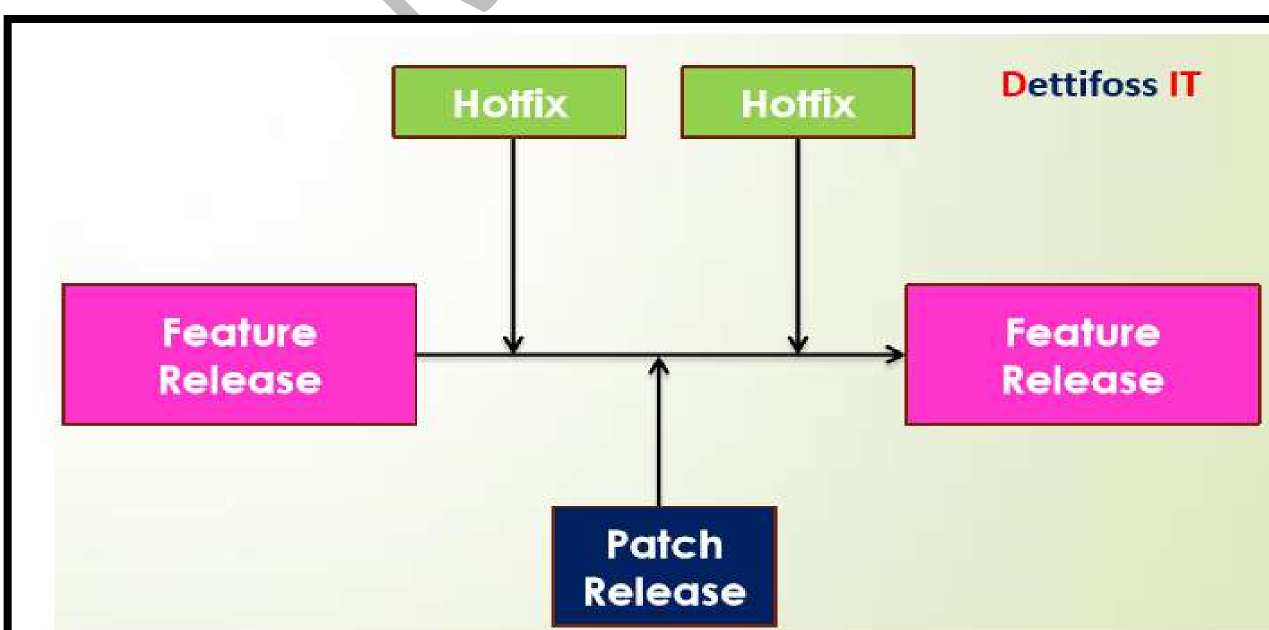
Releases of Service Now

- ✓ Every ~10-12-month release cycle
- ✓ Feature release, patch release, hotfix
- ✓ New UI, new apps, new features

Versions of Service Now

Version Name	Year of Release
Aspen	2011
Berlin	2012
Calgary	2013 Dettifoss IT
Dublin	2013
Eureka	2014
Fuji	2015
Geneva	2015
Helsinki	2016
Istanbul	2016
Jakarta	2017
Kingston	2018
London	2018
Madrid	2019
New York	2019
Orlando	2020
Paris	2020

Releases & Hotfix



Major Customers of Service Now

Service Now User Interface

Different roles in Service Now will experience different UI components.

- ✓ End User
- ✓ ITIL User (Fulfiller)
- ✓ System Administrator

Benefits of Service Now

- ✓ Hosted in the Cloud
- ✓ Easy upgrade
- ✓ Based on ITIL
- ✓ Customization Possibilities
- ✓ Data Confidentiality
- ✓ More Integration
- ✓ Improved Operational Tracking
- ✓ On demand of IT service Management
- ✓ Powerful Reporting Feature

Why is Service Now

1. Reduction of ITSM costs by **80%**
2. Uses Technology as a tool not a barrier
3. Process improvements with agility and low risk
4. Don't have to choose between upgrading configuring

How to use for Business Prospective

1. **Asset and Configuration**: identify and monitor IT service assets and their relationships.
2. **Planned and Policy:** define IT strategies' and manage projects.
3. **IT Services:** deliver IT services and support to business users.
4. **IT Operations Management:** track and manage IT resources and systems.
5. **Non-IT Services:** automate business process outside of IT

Who use Service Now

Employees

To request IT business services.

IT Support staff

To manage service requests, Incidents and problems

Administrators

To manage applications available to selected users, granting them roles and other access controls

Implementers

To deploy process applications and platform features that address an organization business needs

Developers

To create new functionality with scripts or custom applications to extend standard configuration

Who can learn Service Now

- ✓ There are no prerequisites to learn service now
- ✓ Although awareness of programming knowledge would be little helpful during the course
- ✓ Every API feature used, will be based on the service now platform

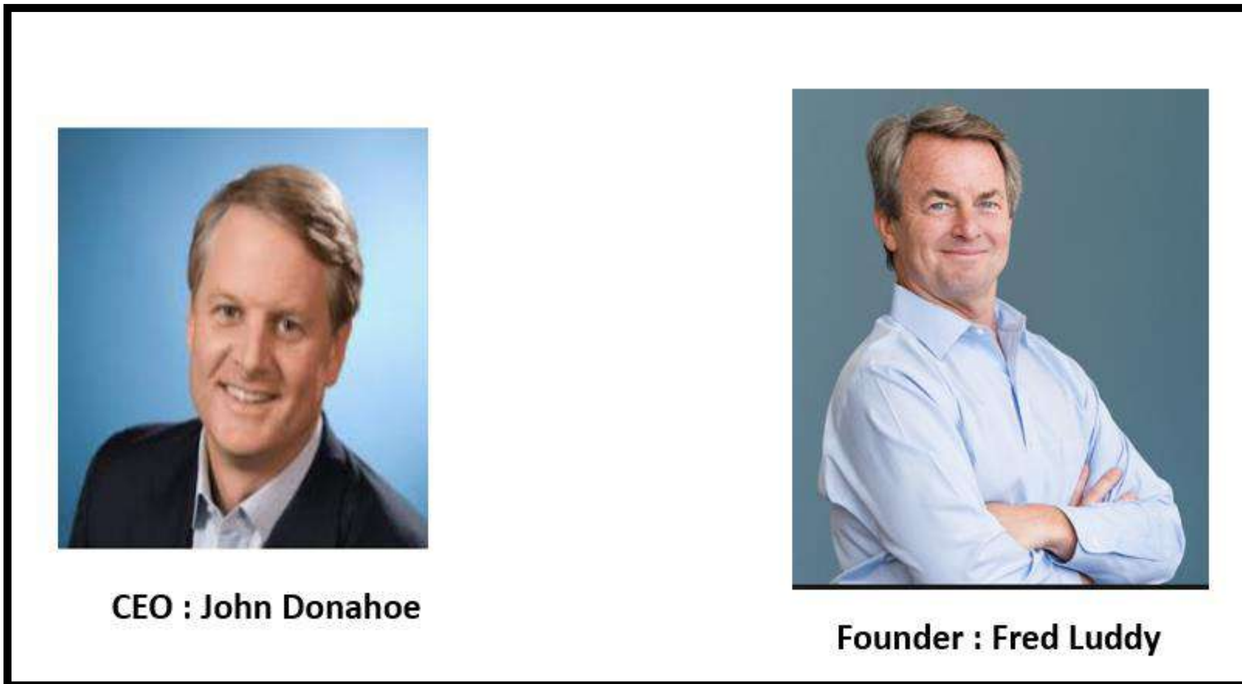
History of Service Now

- ✓ Service Now, Inc. is a cloud computing company headquartered in Santa Clara, California.
- ✓ It was founded in 2004 by **Fred Leddy**
- ✓ The previous CTO of software companies Peregrine Systems and Remedy Corporation.
- ✓ Service Now is listed on the New York Stock Exchange and is a constituent of the Russell 1000 index
- ✓ This tool is based on the ITIL Standards

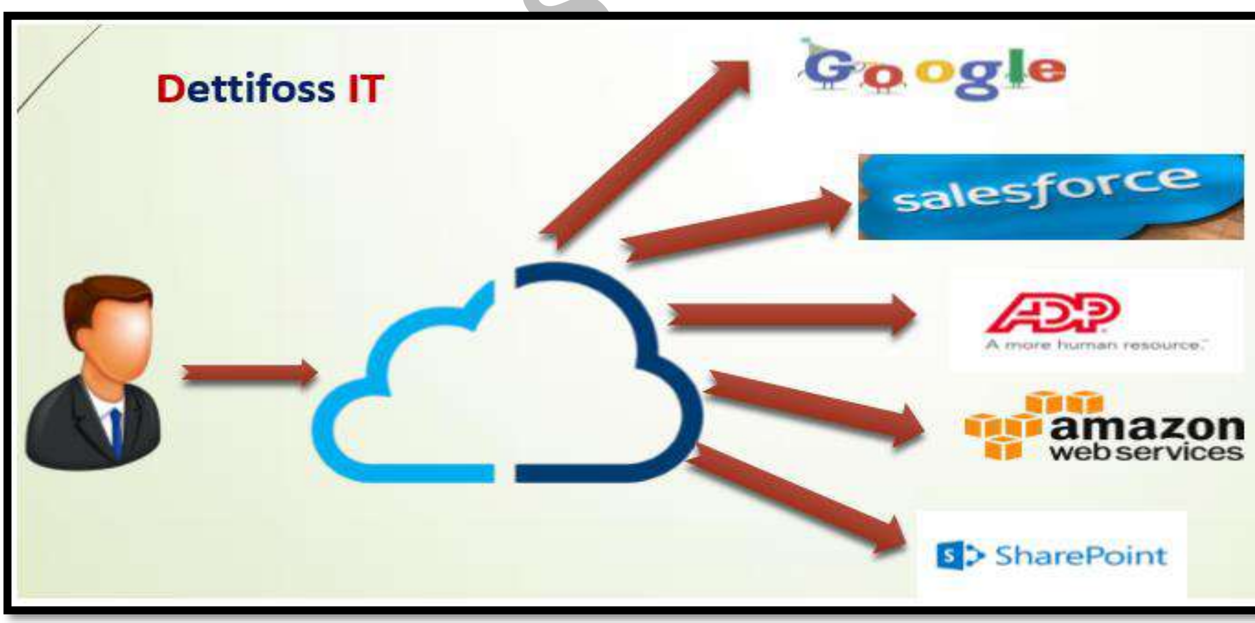
Features of Service Now

- ✓ Asset management.
- ✓ Change and release management.
- ✓ Configuration management.
- ✓ Incident management.
- ✓ IT cost management.
- ✓ Problem management.
- ✓ Service catalog.
- ✓ Demand management.
- ✓ Simple and Consistent
- ✓ Flexible
- ✓ Secure
- ✓ Easy to Integrate
- ✓ Speed time to Production
- ✓ All is in Table format
- ✓ Different type of Modules
- ✓ Different Types of Users
- ✓ Update Set
- ✓ Functionality in Service now
- ✓ Manage the Data (group, users)
- ✓ Load the Data
- ✓ Import the data
- ✓ Export the data
- ✓ Email Notification and SM
- ✓ Workflow
- ✓ Reporting
- ✓ Mobile, Desktop, Tablets view
- ✓ Live feed (Chat Portal)
- ✓ Catalog
- ✓ Knowledge Module
- ✓ Service Portal
- ✓ Content Management System (CMS

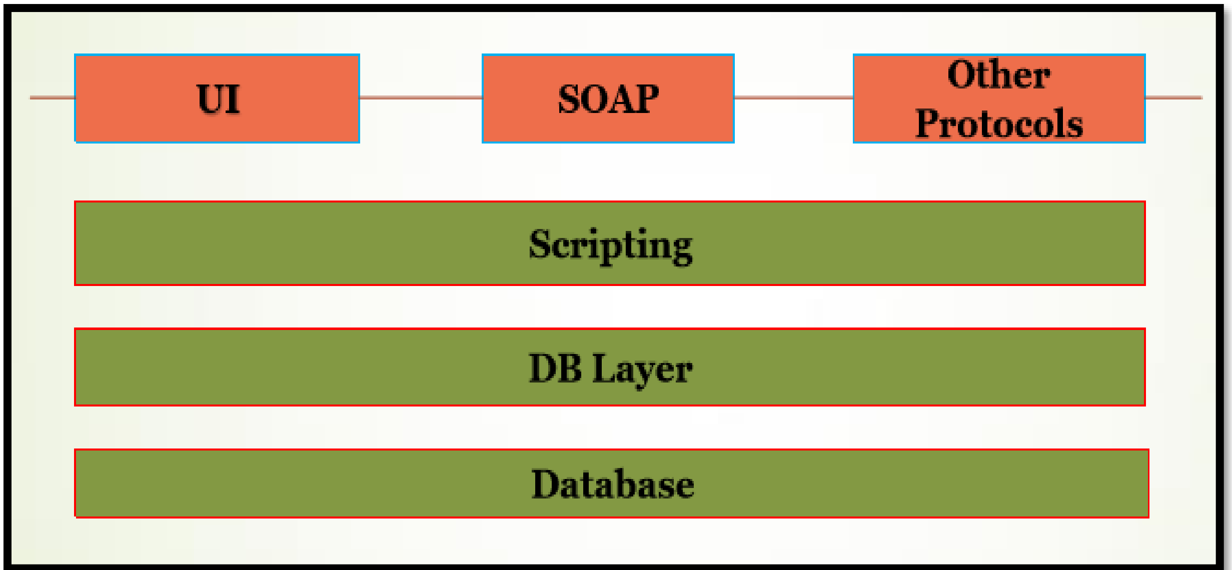
Founder and CEO Service Now



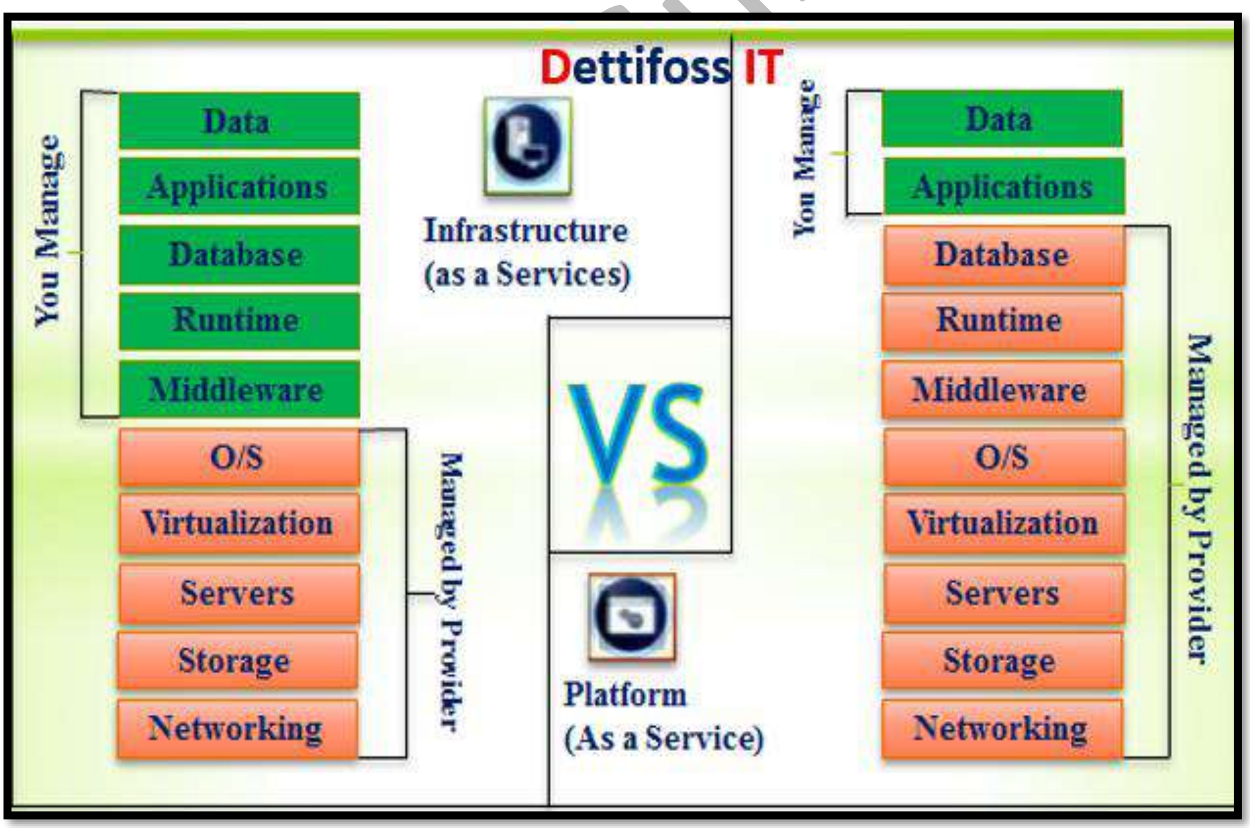
What is SaaS



Service Now Architecture



Iaas Vs. Paas



Chapter - 2

Personal Instance Account Creation

Service Now Developer Account Creation

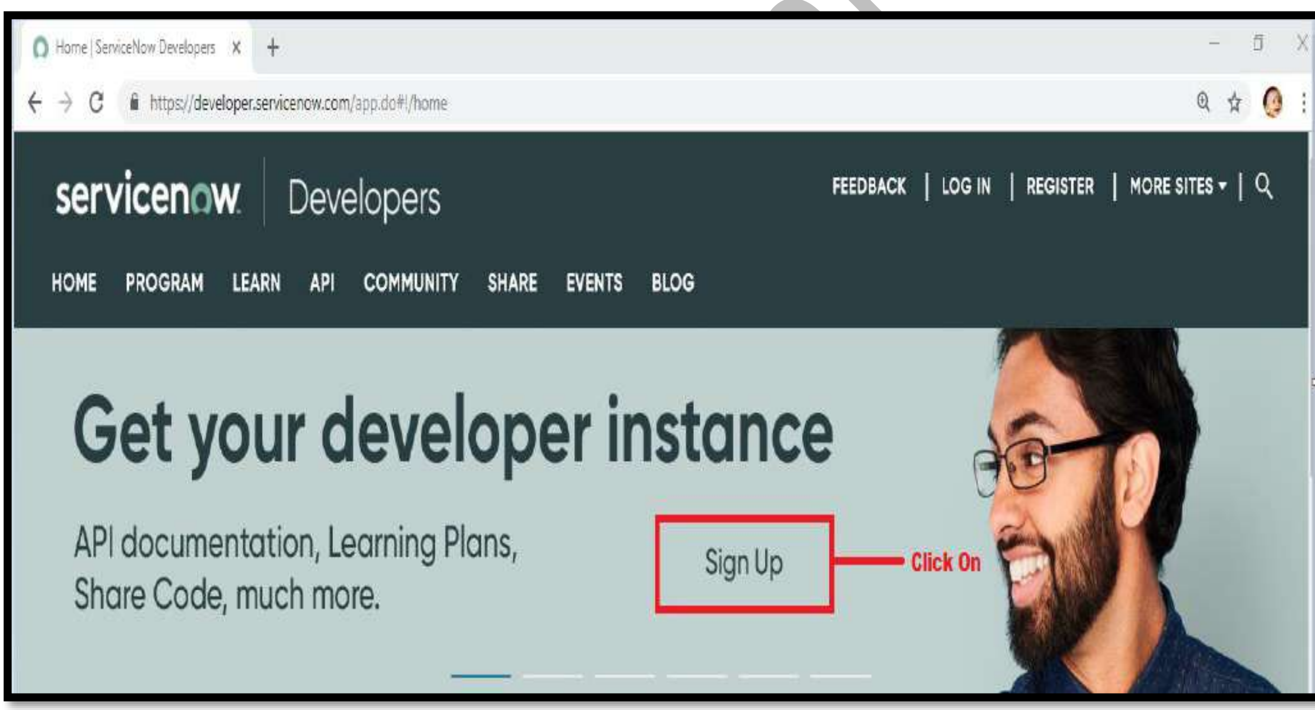
In this chapter we will learn how to create a personal developer instance:

Service Now offering free and full-fledged personal developer instances **(PDI)** to registered users who want to develop scope applications on the Service Now platform or improve their skills with Service Now. Members of the Service Now Developer Program can use their PDI as long as there is activity on the instance.

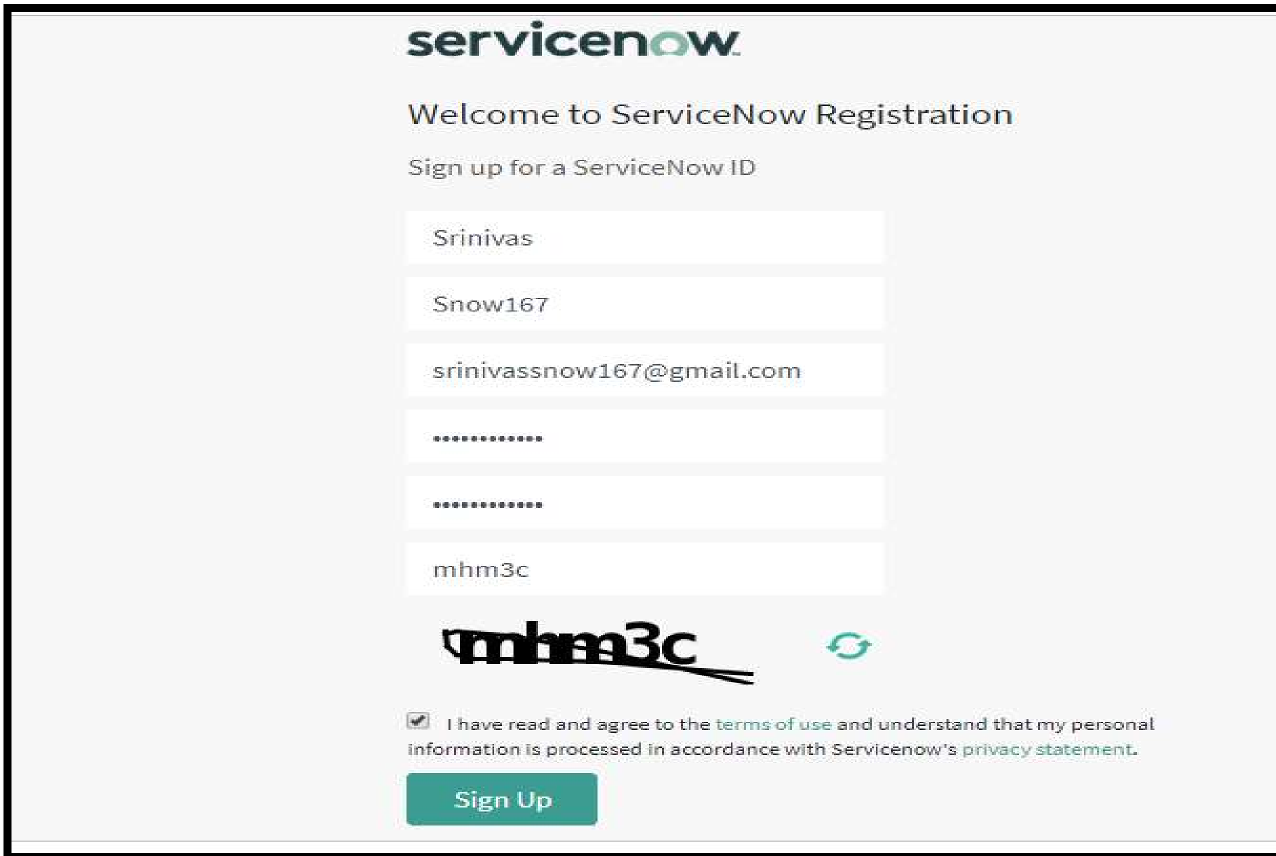
Exercise 1: Creating a Personal Developer Instance

Create a personal developer instance is very fast and easy.

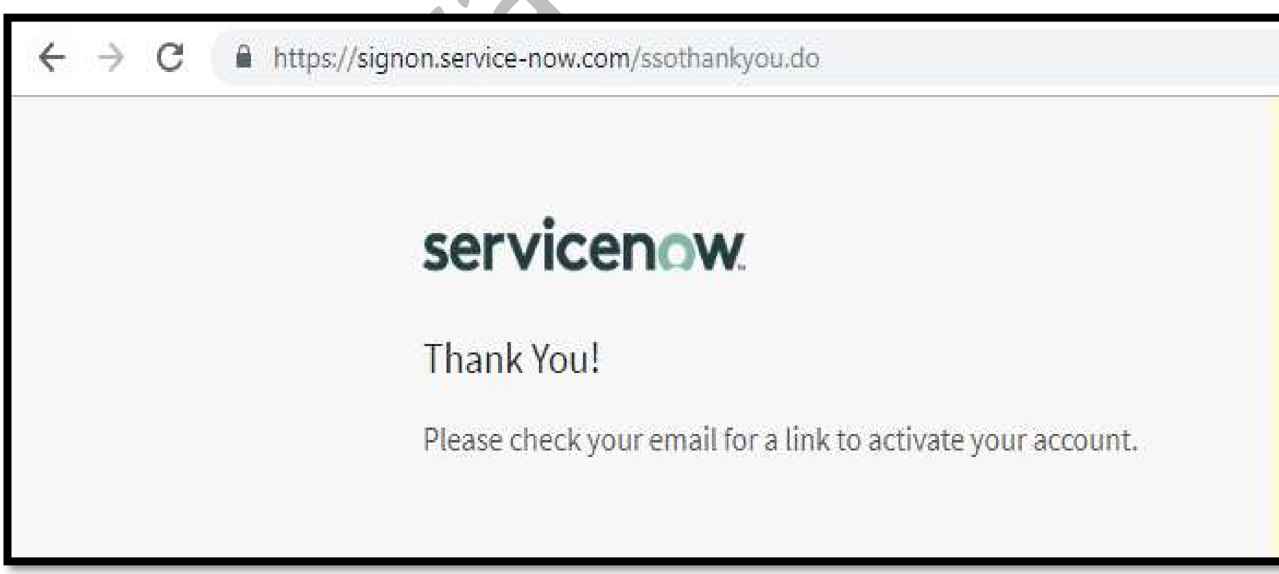
1. Log in to the **Service Now developer site (https://developer.servicenow.com)**



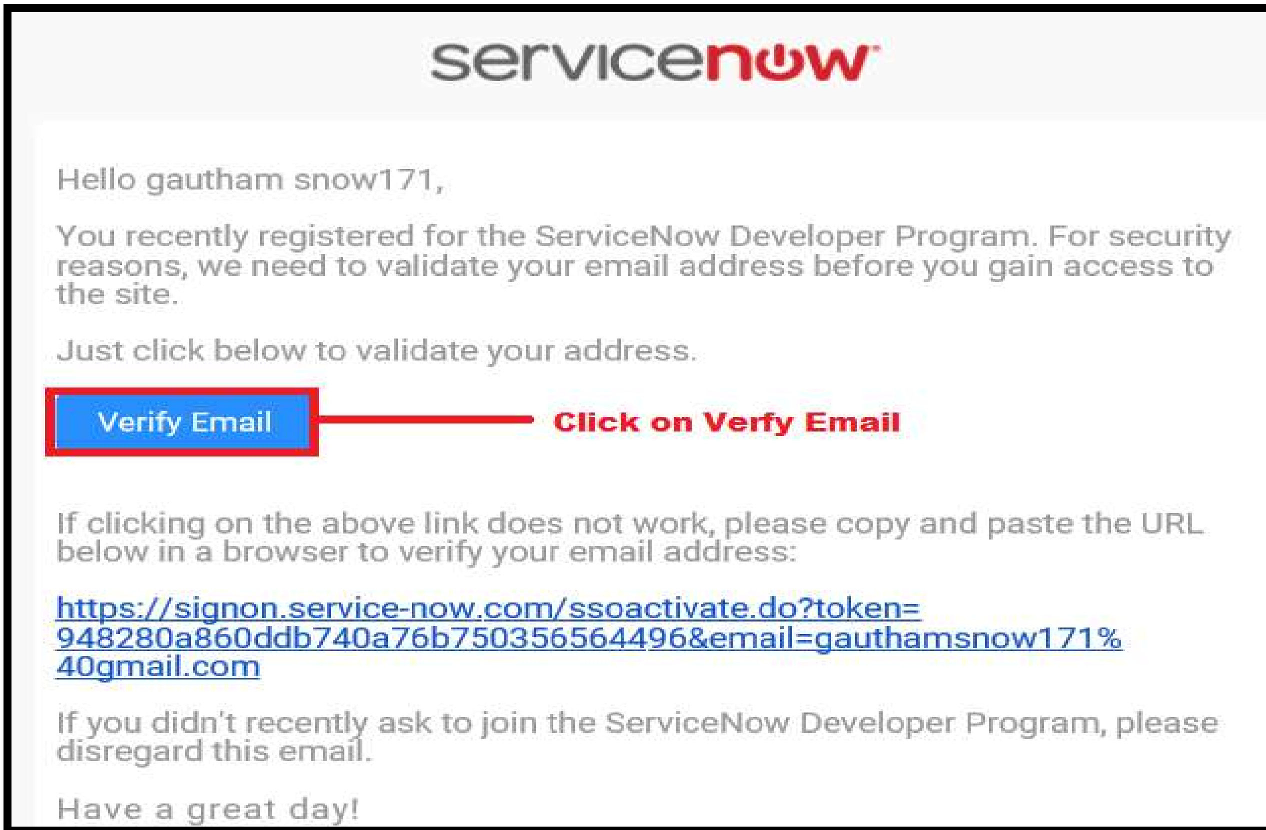
2. Provide all **required** details below



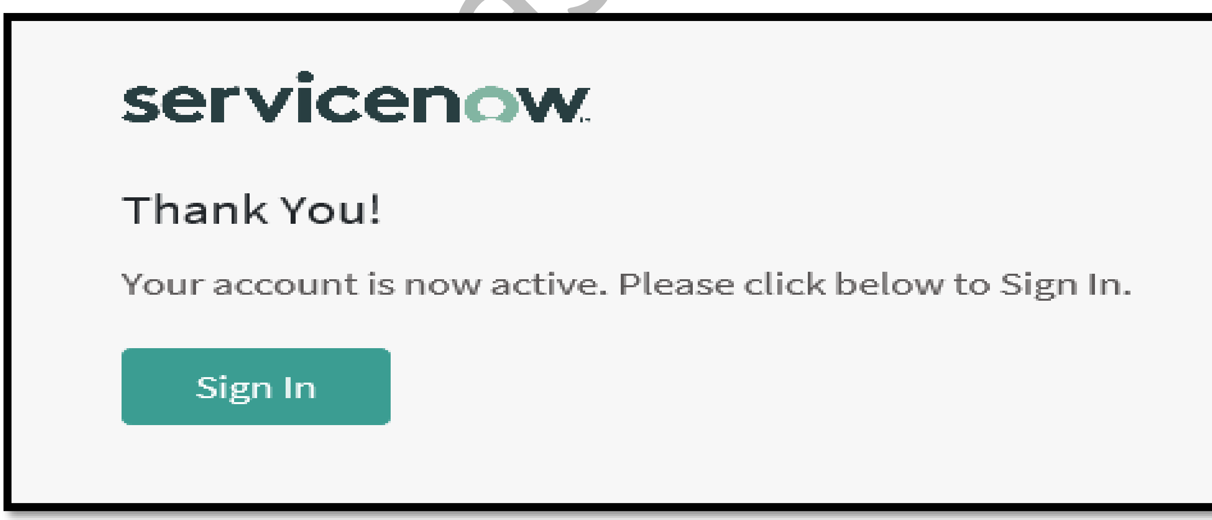
3. Click on **Sign Up**



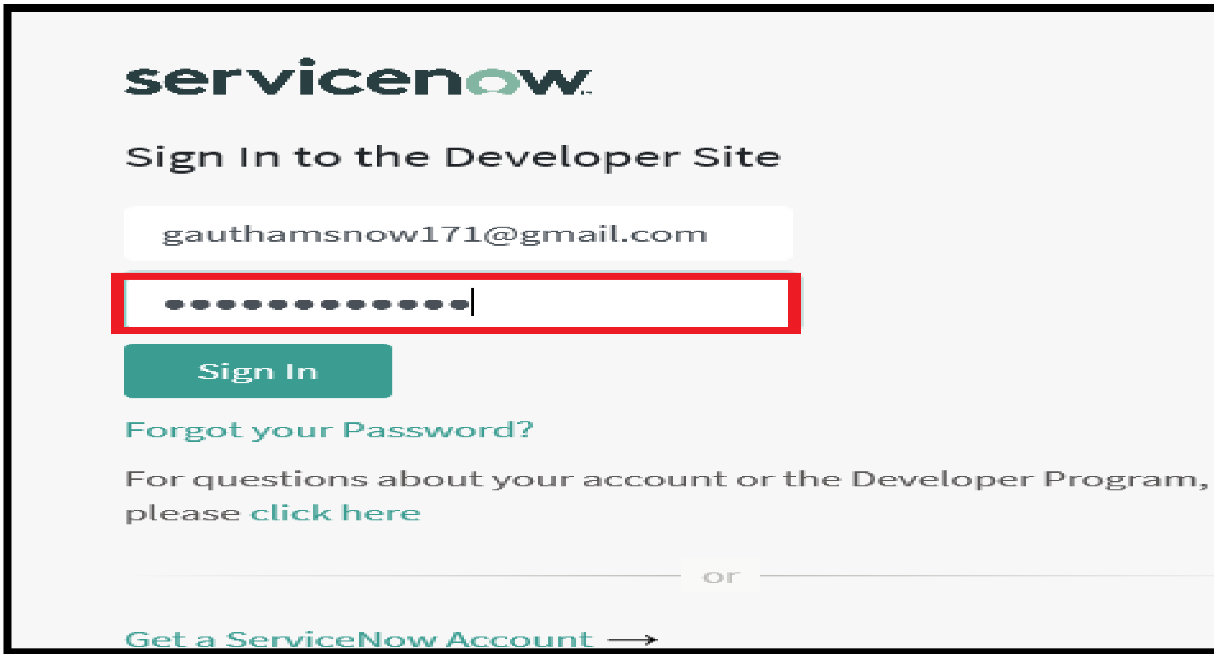
4. Open your **Gmail account** and verify it



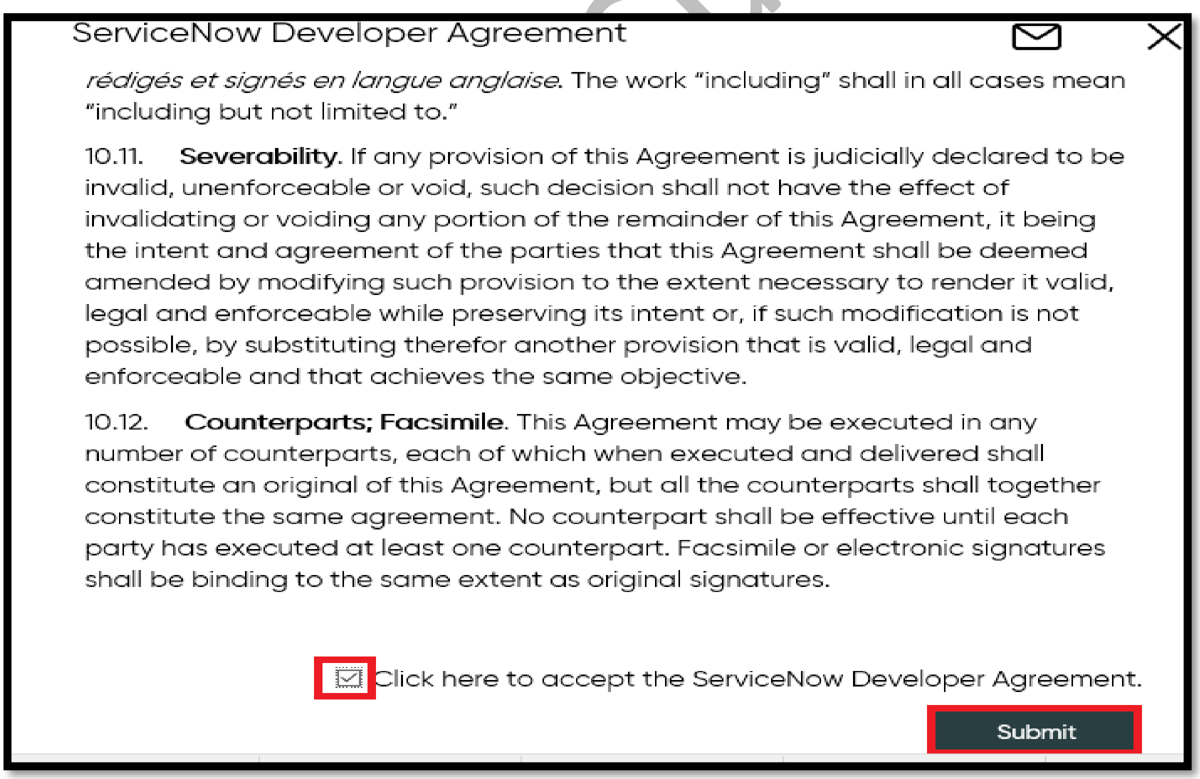
5. Now account is active, **try to login in account**



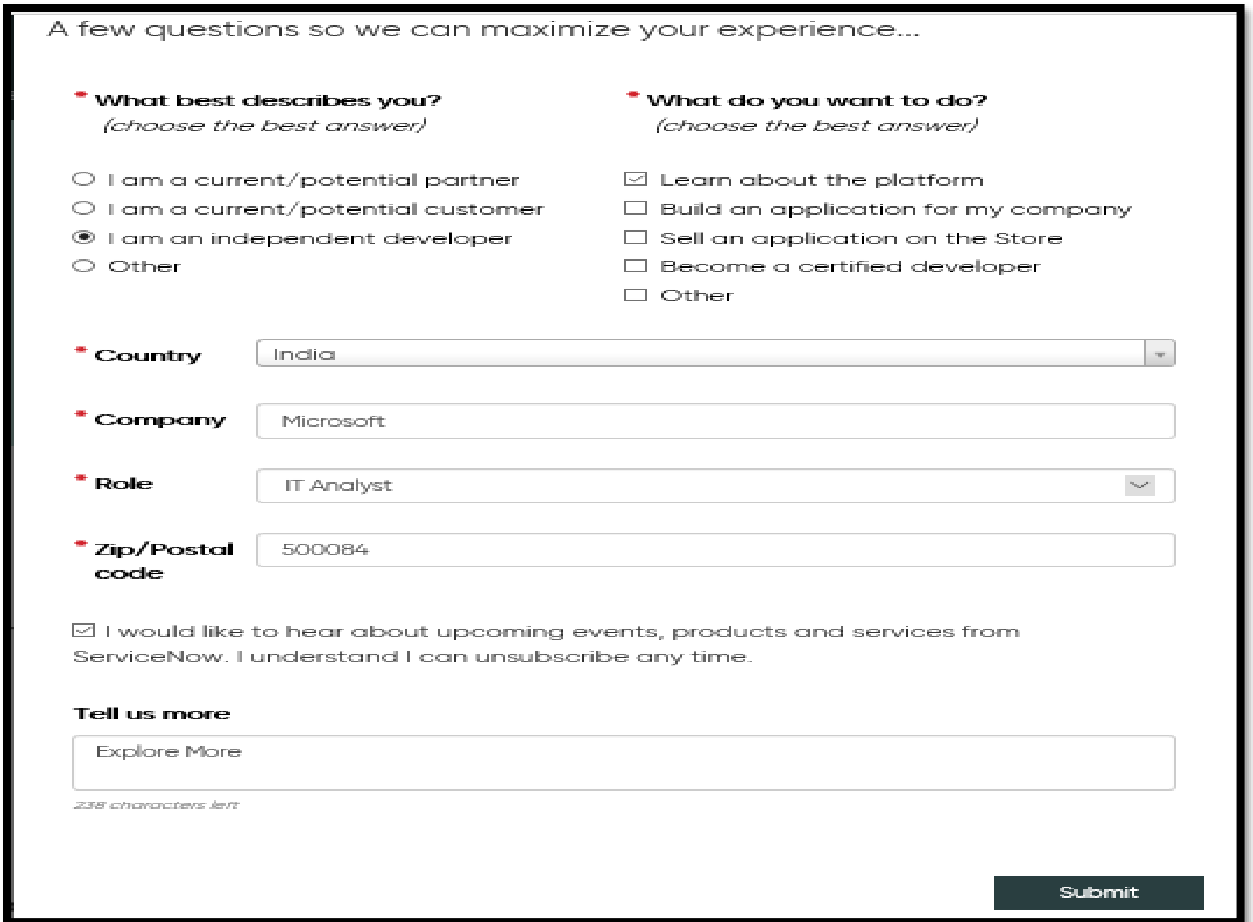
1. Click on **Sign in** and Provide your account password and



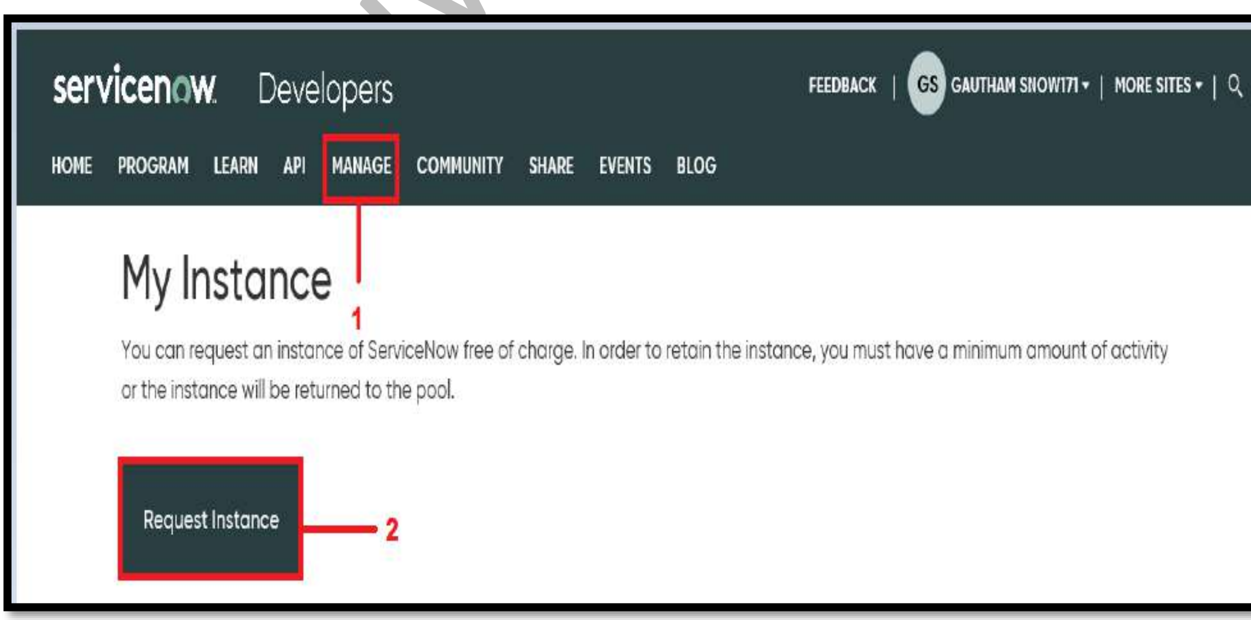
2. Sign on developer agreement and click on **Submit**



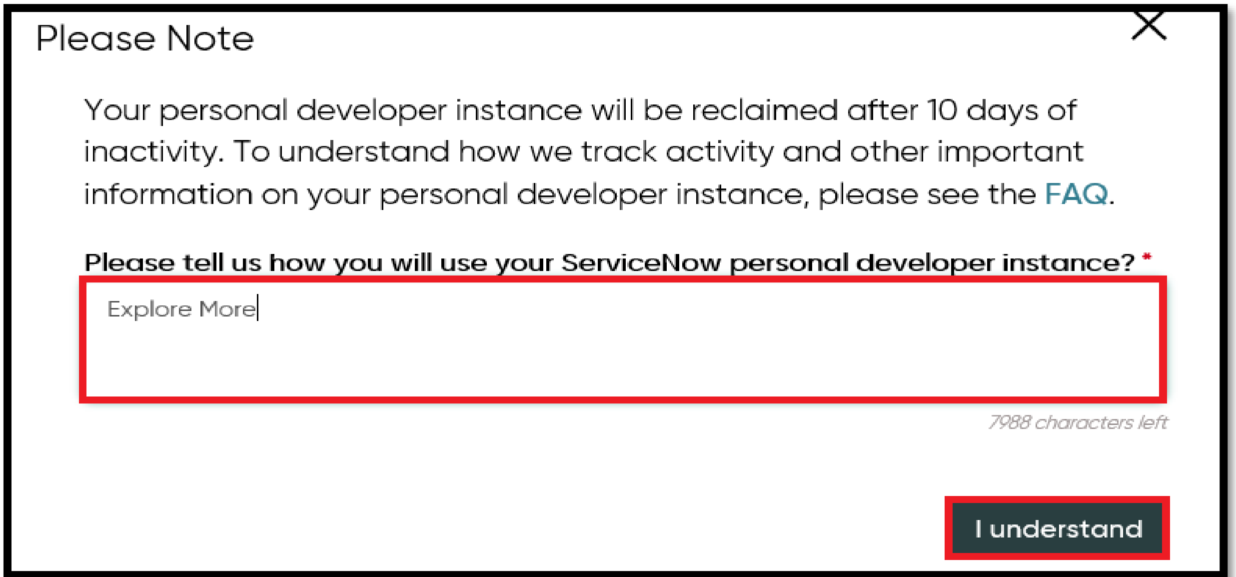
3. Fill all mandatory fields on below form and click on **Submit**



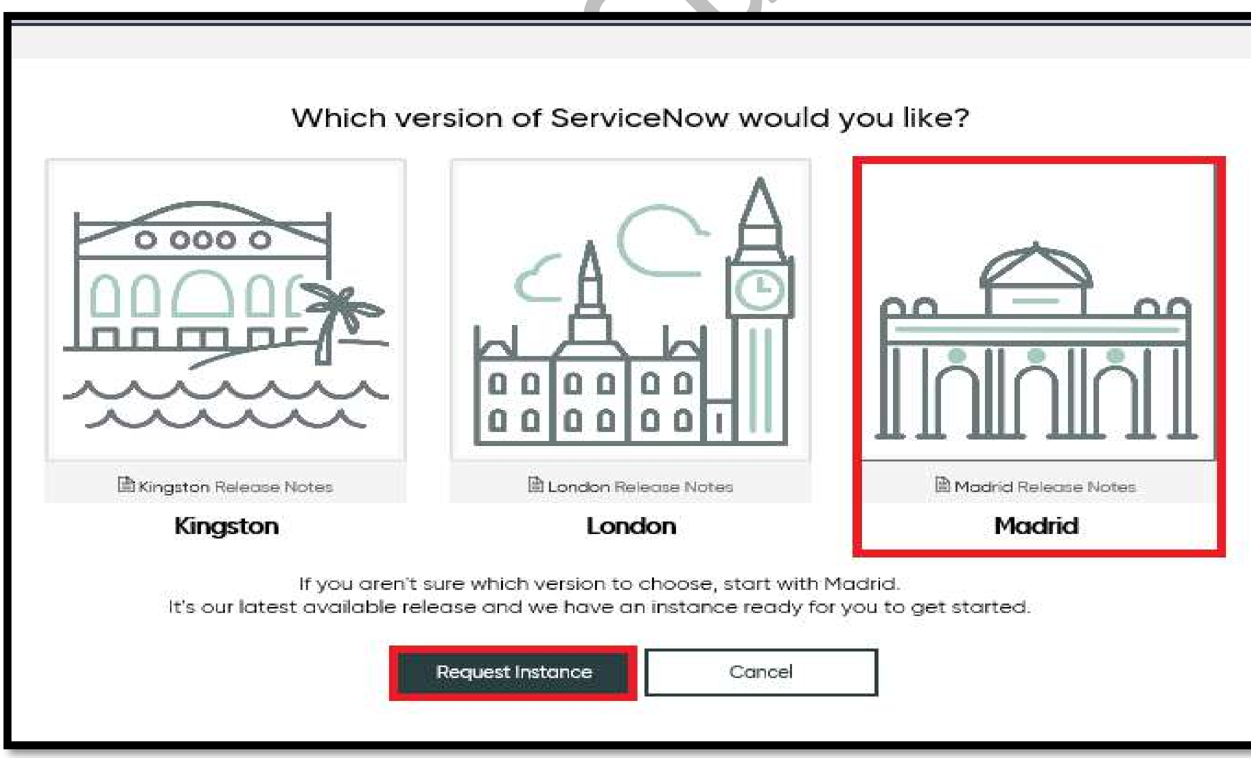
4. Click on **Manage Menu** and click on **Request Instance** button



5. Fill note field and click on **I understand** button

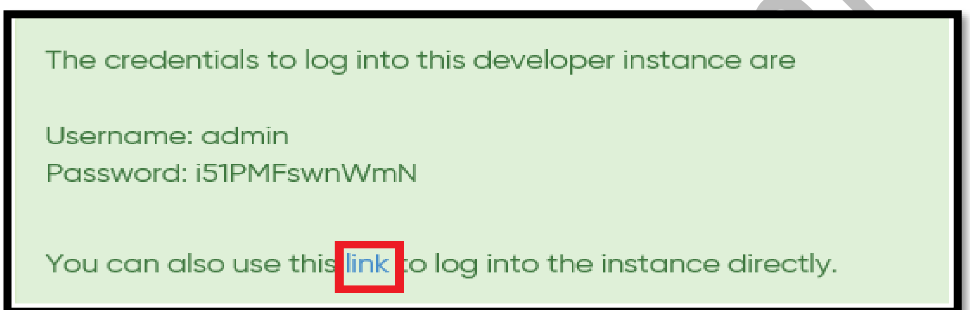


6. Select a Service Now **Version** for the instance.
7. Service Now developer instance offering **latest 3 versions**, in this case I have selected **Madrid**

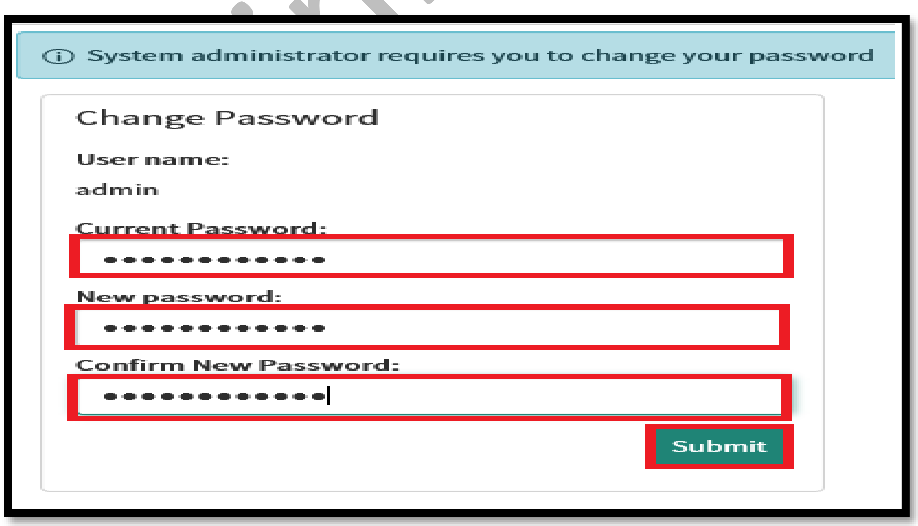


8. Click on **Request Instance** button
9. Once created an account it will assign one domain id for each individual personal developer instance and provide credential details

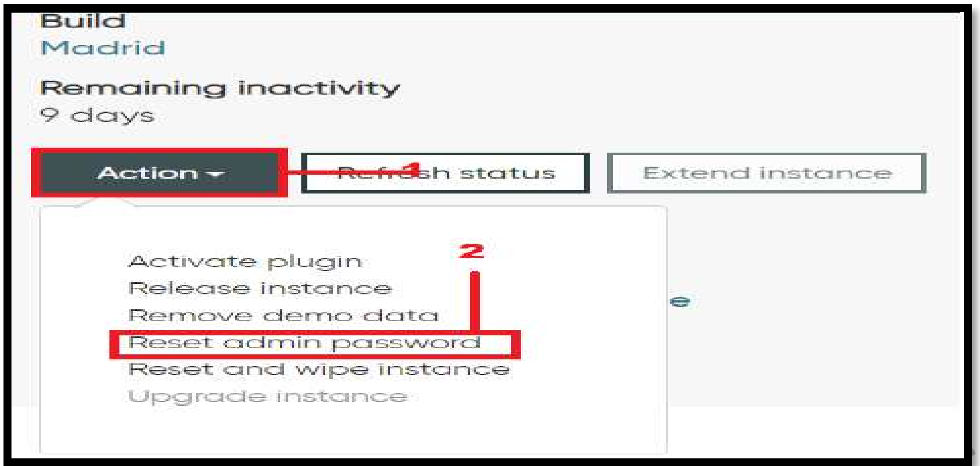
10. Remember **username** and **copy password** then click on link



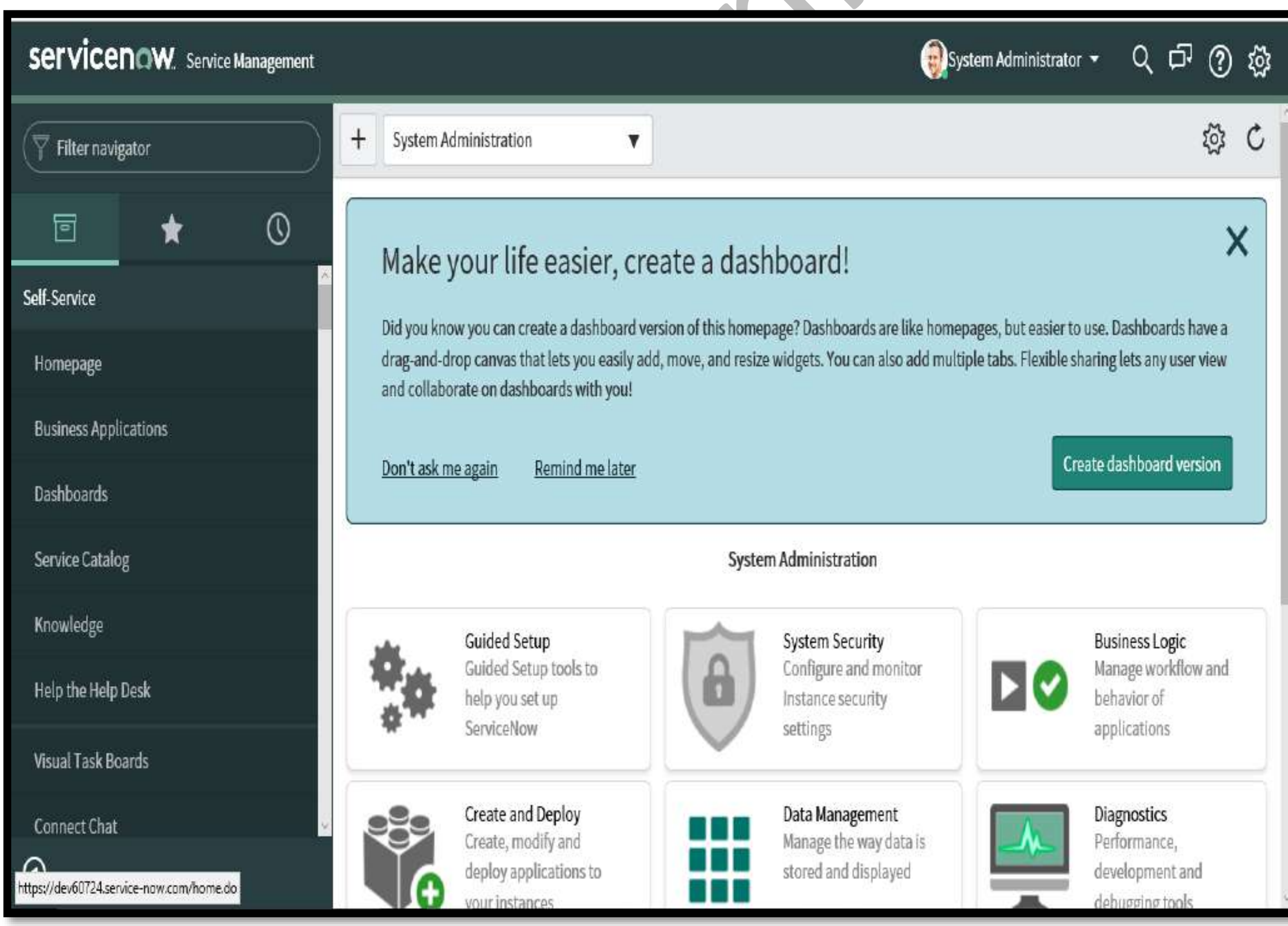
11. Past our password into **current password** field and set **new password and confirm new password**



12. Click on **Submit** button



Note: If you navigate away from the My Instance page before copying the admin credentials, you will need to reset the admin password. Click the **Action Button** and select the **Reset admin password menu** item. Follow the prompts to get a new admin password.



18. Service Now **Home Page** Look like above screen shot

In order to keep the developer program free and provide **PDIs** to everyone who wants one, instances hibernate when they are idle. You can wake up a hibernating instance by navigating to **Manage > Instance** on the **Service Now developer site**. It takes approximately three minutes to wake up a hibernating instance. Your work is saved on hibernating instances.

If there is no instance activity within **10 days**, the instance is reclaimed. Reclaimed instances and their data cannot be recovered. Log in to your instance and use it regularly to keep the instance active.

Exercise 2: Login into Personal Developer Instance

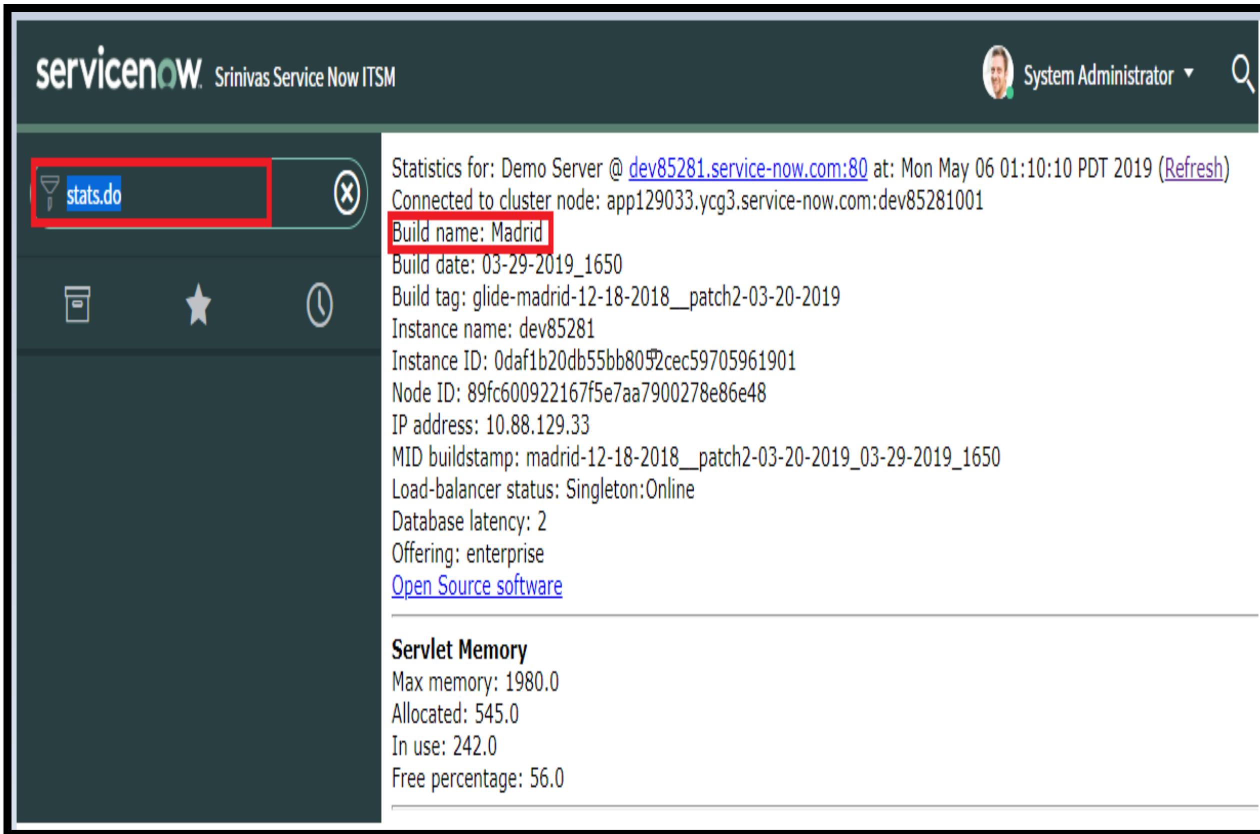
In this exercise, we will learn how log in to a personal developer instance (PDI).

1. In a web browser, open **developer.servicenow.com.**
2. Log in with your developer site **username** and **password**. If you are not currently a member of the Service Now developer site, create an account.
3. **Open Manage > Instance.**
4. Click the **Request Instance** button.
5. Select Madrid as the version. If Madrid is not available, select any available previous version. After the instance is spun up, use the Actions button to upgrade to the version of your choice.

It is not possible to downgrade versions. When your instance is available, you will see the Instance URL and credentials.

How we come to know which version currently using

1. **Move to Filter Navigator >** Open application **Stats.do**
2. Click on **Enter**


servicenow. Srinivas Service Now ITSM
System Administrator
stats.do
Statistics for: Demo Server @ dev85281.service-now.com:80 at: Mon May 06 01:10:10 PDT 2019 (Refresh)
Connected to cluster node: app129033.ycg3.service-now.com:dev85281001
Build name: Madrid
Build date: 03-29-2019_1650
Build tag: glide-madrid-12-18-2018__patch2-03-20-2019
Instance name: dev85281
Instance ID: 0daf1b20db55bb8052cec59705961901
Node ID: 89fc600922167f5e7aa7900278e86e48
IP address: 10.88.129.33
MID buildstamp: madrid-12-18-2018__patch2-03-20-2019_03-29-2019_1650
Load-balancer status: Singleton:Online
Database latency: 2
Offering: enterprise
Open Source software
Servlet Memory
Max memory: 1980.0
Allocated: 545.0
In use: 242.0
Free percentage: 56.0

Chapter-3

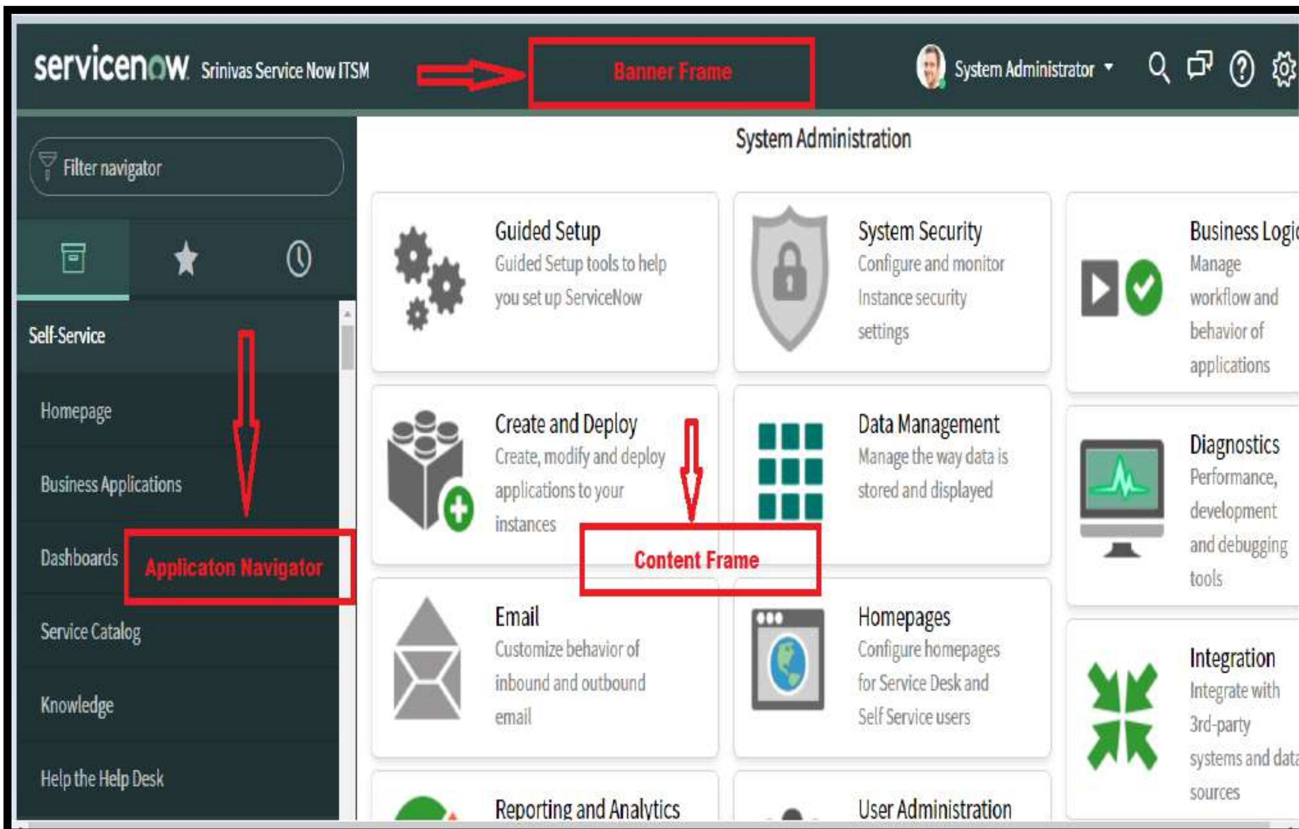
Service Now User Interface

Service Now User Interface

In this chapter we will obtain about service now user interface:

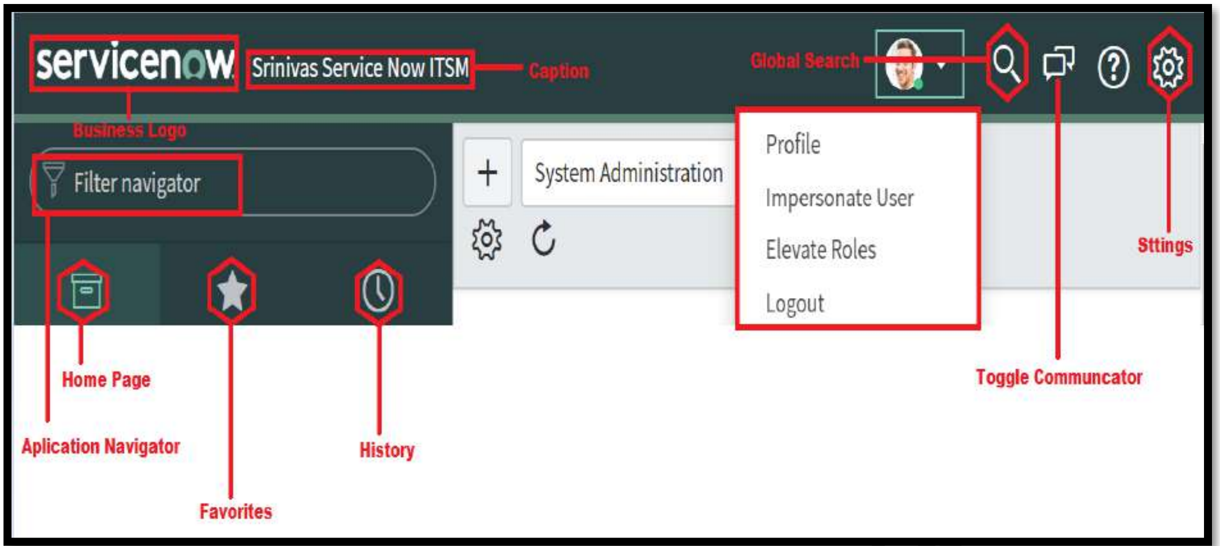
There are three components of the Service Now user interface **UI16**:

1. Banner Frame
2. Application Navigator **or** Left Navigator **or** Filter Navigator
3. Content Frame



Working with Banner Frame

The banner frame shown the top of every service now home page. It contains a **logo image and global navigation controls**. The controls are restricted based on user **role**. The admin user sees all of the controls. The *Settings* control ( ) contains a pane for setting developer options. The banner shows which user is logged in.



The Banner Frame of UI16 contains the following controls:

1. **User Menu:** This control allows the user to access and modify **his logged in user profile.**
2. **Impersonate User:** Administrators can impersonate other authenticated users for testing purposes or better understanding customer issue and view impersonation logs. The impersonation option is not visible in the mobile view of the platform.
   by default, it visible to **admins**

   When impersonating another user, the administrator has access to exactly what that user can access in the system, including the same **menus and modules**. The instance records anything the administrator does while impersonating another user as having been done by that user.

   Note: Impersonate roles is applicable to only particular session,

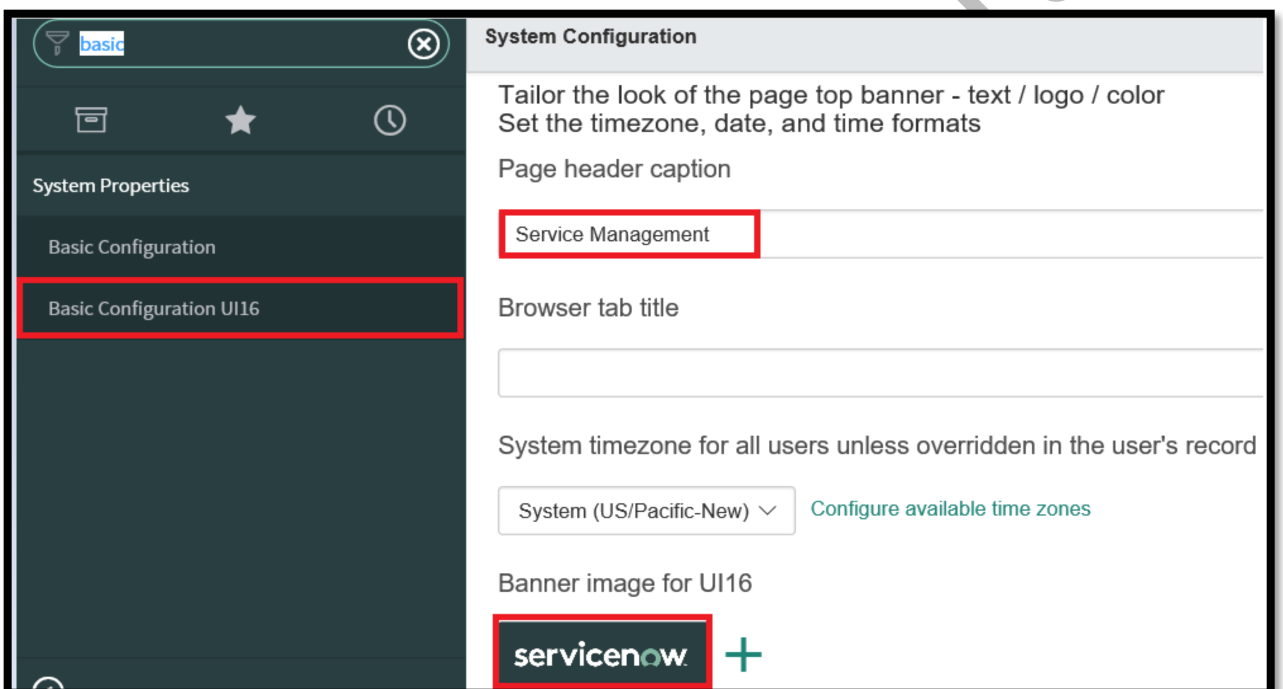
3. **Elevated Privilege Roles**: It require you to manually accept the responsibility of using the role before you can access the features of the role.
4. **Global Search:** This control helps to search for records from tables
5. **Connect sidebar icon:** Clicking this control opens a tab on the right side to connect with the co-workers and also find existing conversations. This icon is available if Connect (plugin for live feed) is enabled.
6. **Help icon:** Clicking this icon opens a tab on the right side with 3 links that would help the user with documents on service now
7. **Settings icon:** This control opens a popup window with various options for system settings. This window also has a button to switch the UI from UI16 to UI15 or vice versa.

Exercise 1: Configure logo, colors, system defaults for UI16

We can use the **Basic Configuration UI16** module to change brand your instance with your **company logo** and **colors** and set **basic system defaults**. This is the best place to start if you are setting up your instance for the first time or if you have recently enabled UI16.

Procedure

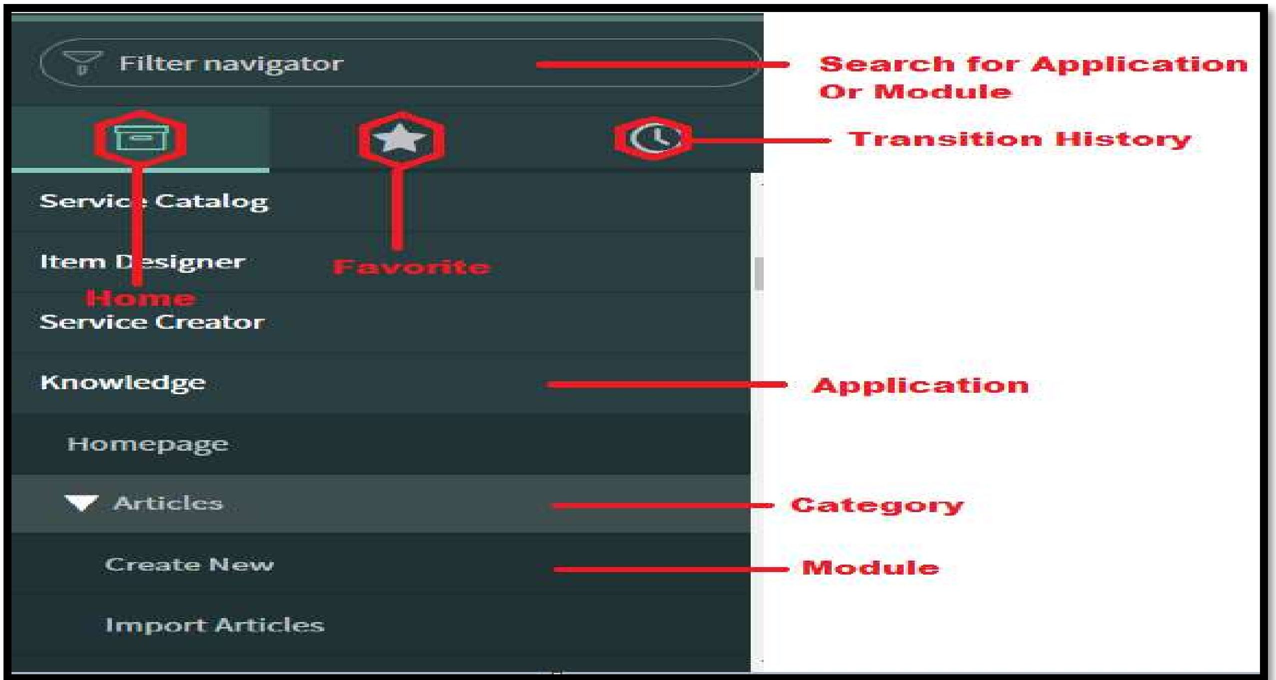
1. Navigate to **System Properties > Basic Configuration UI16**
2. Complete the configuration by changing any of the following settings.
3. Basic system configuration properties



Working with Application Navigator

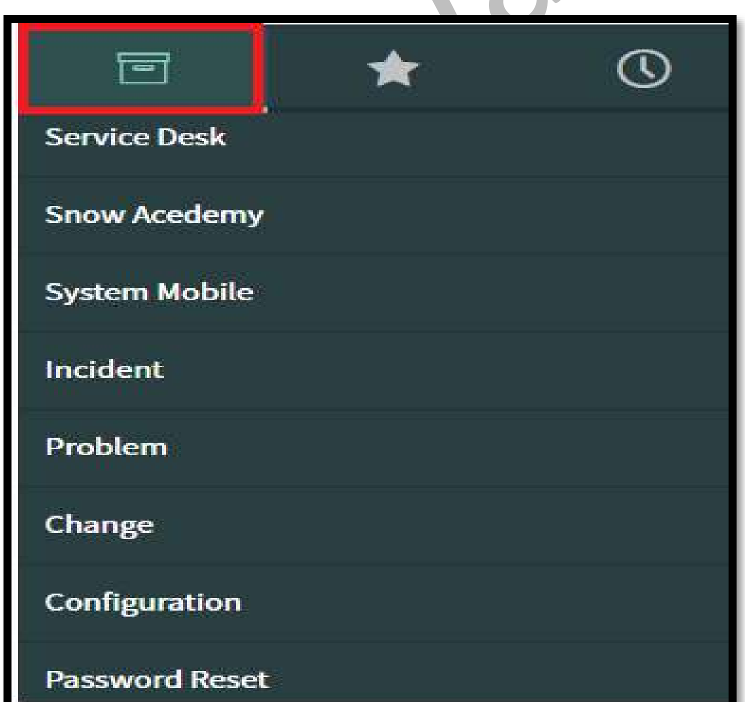
The **application navigator** provides access to all **applications** and the modules they contain, enabling users to quickly find information and services. An **application** a group of modules, or pages, that provide related information and functionality in an instance.

1. Also called **left-navigation** bar
2. Provides links to all applications and modules to quickly find information and services.
3. Each application appears as a section in the application navigator denoted by an application label.
4. Modules are listed by name Under each application label.
5. **UI16 has 3 tabs in** the application navigator



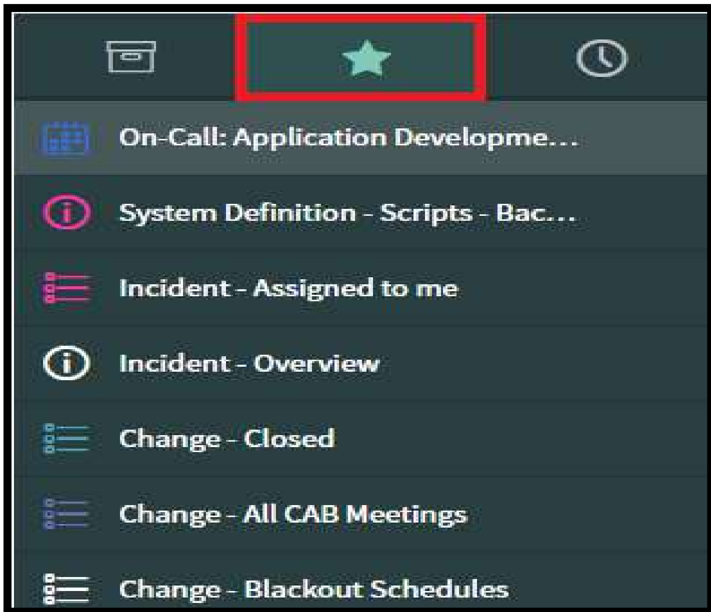
1. All Applications

Displays all application menus and modules. Each application appears as a section in the application navigator denoted by an application label. Modules are listed by name under each application label.



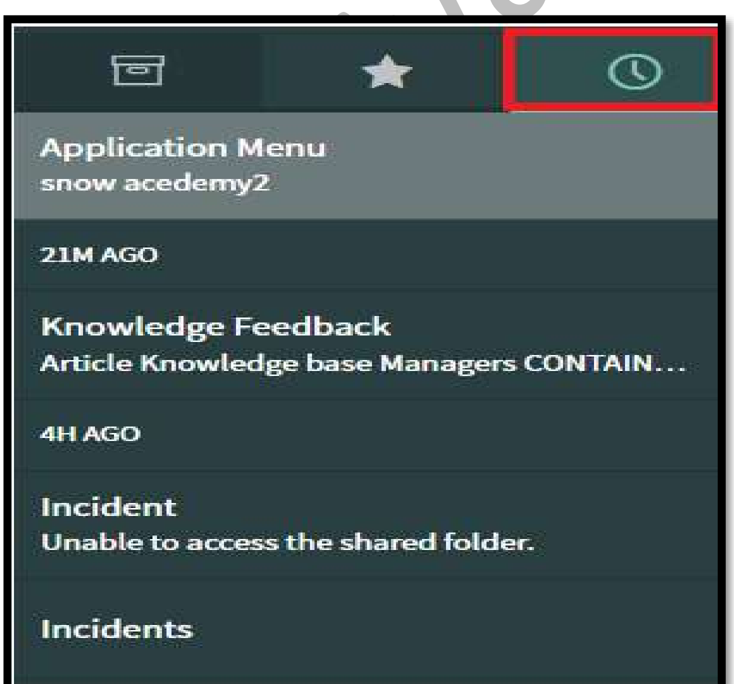
2. Favorites

Displays items you have added as favorites. If you upgrade to UI16 from a previous interface, any **bookmarks** you created are automatically converted to favorites.



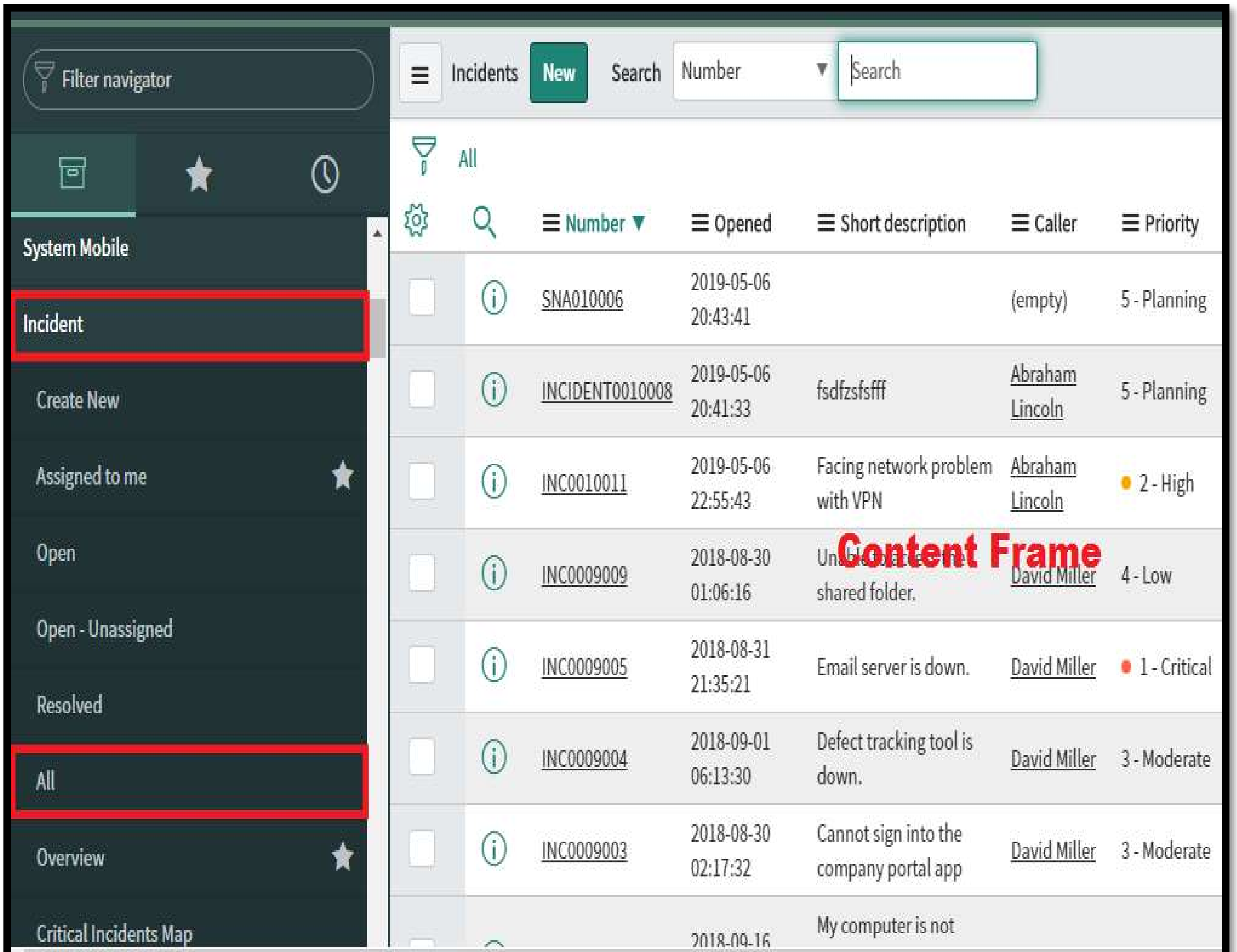
3. History

Displays transitions' you have recently accessed



Working with Content Frame

The content frame displays content pages when users select modules in the Application Navigator. The pages in the content frame render data in many ways. The most basic page types are lists and forms.

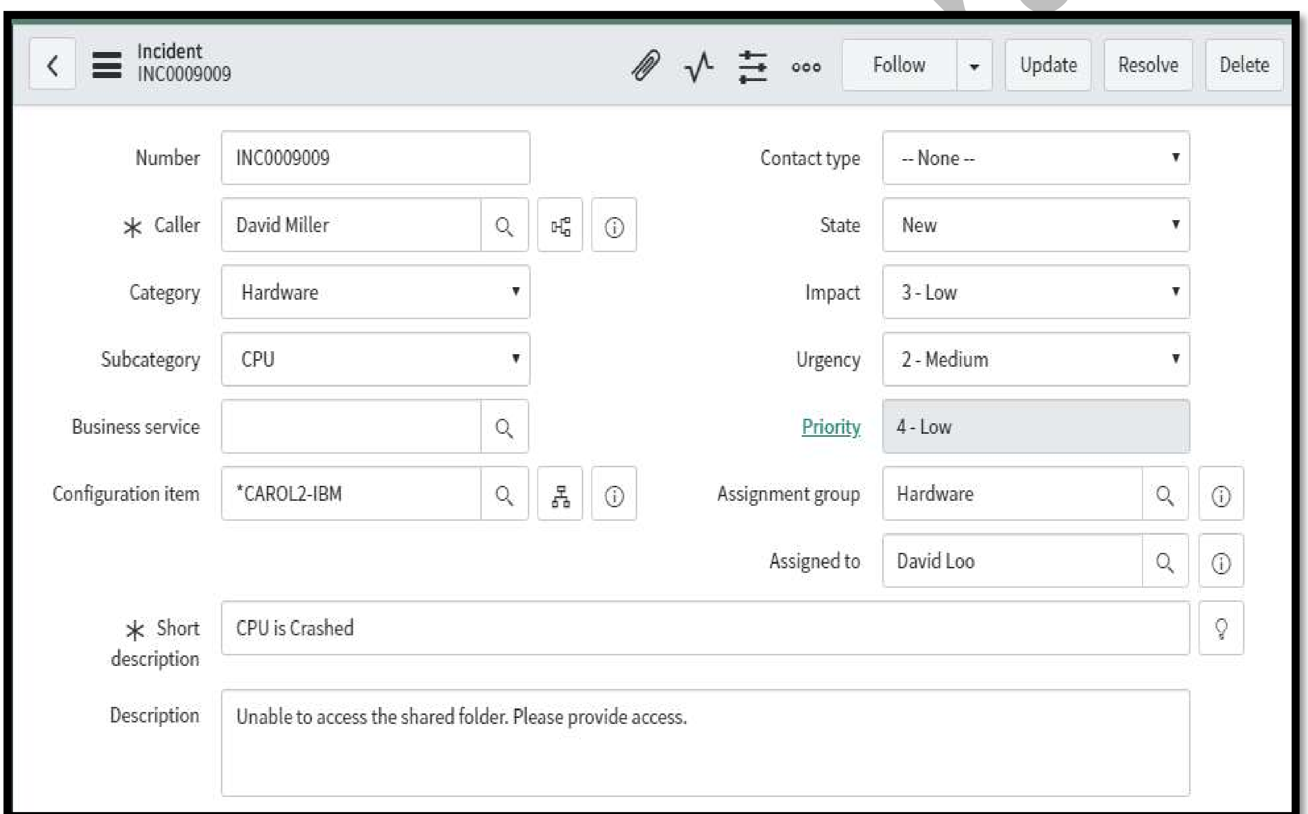


Chapter-4

Forms

What is Form (Service Now)

1. A form **displays information** from one record in a data table.
2. The specific information on a form depends on the **type of record** displayed.
3. Users can **view and edit** records in forms.
4. **Administrators** can configure what appears on forms.
5. A form is a content page displaying fields and values for a single record from a database table.
6. Forms have a **1-column layout, a 2-column layout**, or a mix of both.
7. Forms are opened from modules in the Application Navigator or by clicking a record's number in a list.

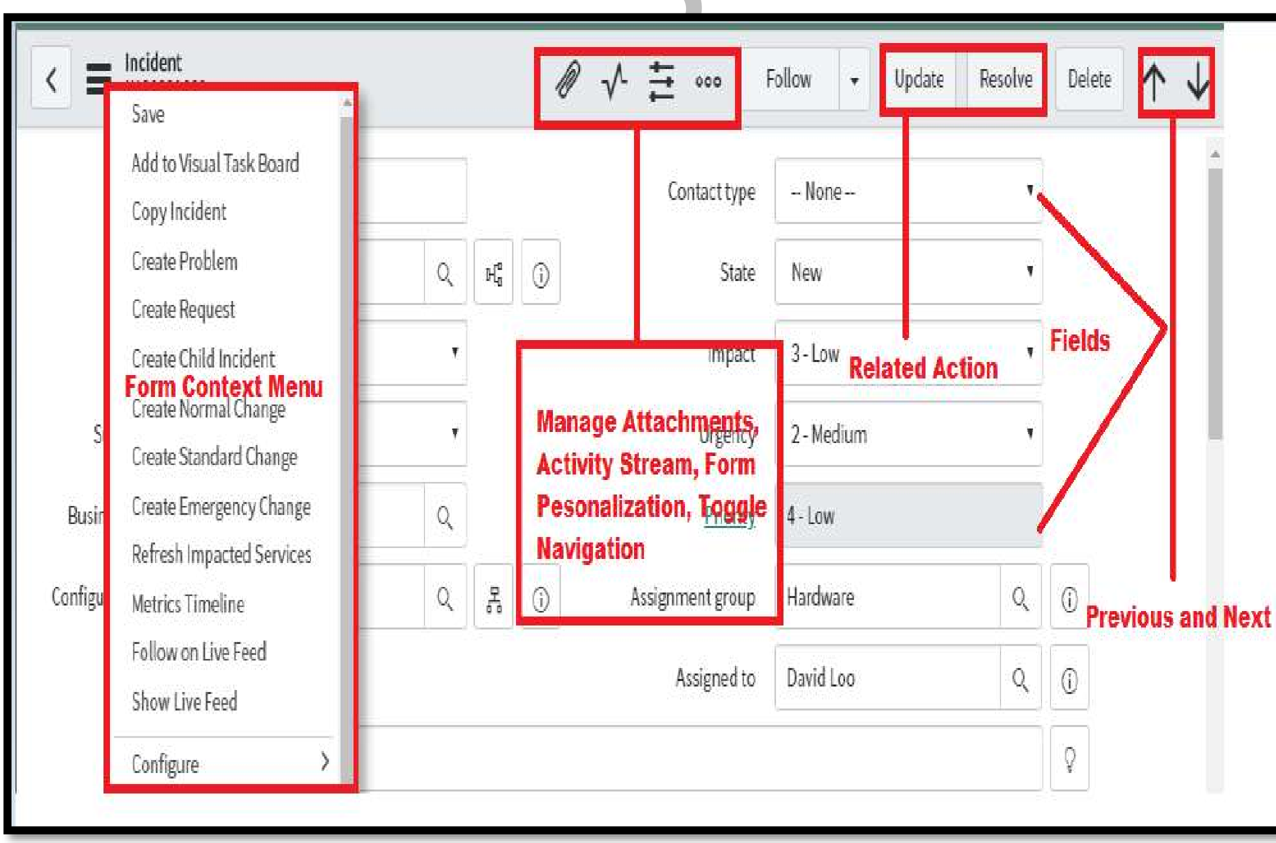


What is Record

1. A single entity
2. Each record has a unique key

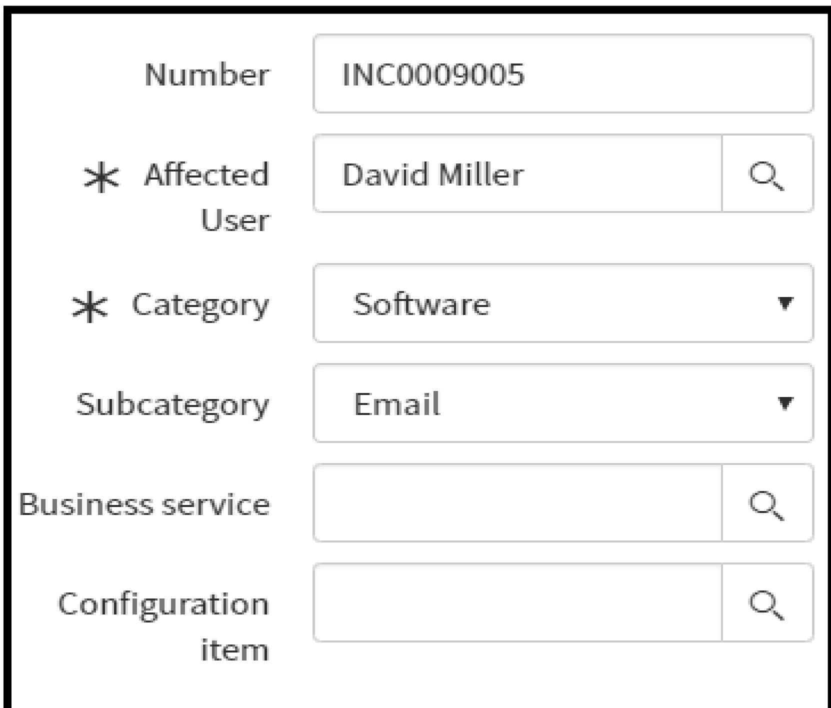
Forms and Form Headers

1. Single Record
2. Many attributes
3. Customize form
4. The form header displays the title of the form (table) and provides several controls
5. The UI16 form header includes the following control



Forms Fields

1. A field represents an **individual item** of data on a record
2. Users can view and modify field data on a form. For more detailed information,
3. The individual pieces of data in a record are called fields.
4. Users enter data in fields on the form or by using the list editor.
5. Administrators can create new or modify existing fields
6. Users can enter data in fields by using the list editor or by using a form.
7. In form view, fields appear as fields in the form, and in list view they appear as columns of data in the table.

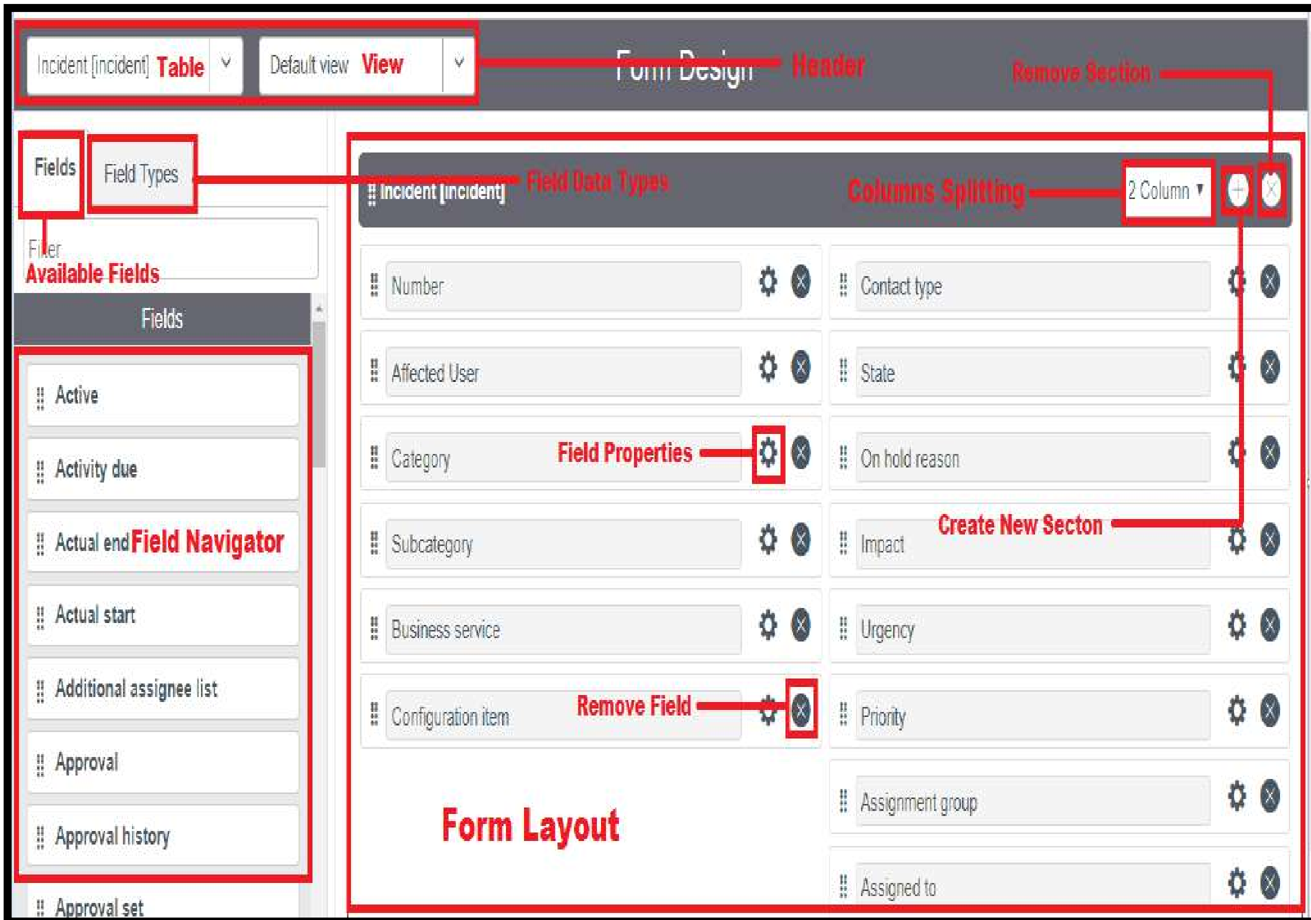


Configure Form Using the form designer

Administrators or users with the personalize form role can use the form design feature to quickly create new or change existing form views.

Form design is an alternative to configuring forms that combines several configuration options into one tool. It is available by default for new instances and requires **UI16 or UI15.** Administrators who upgrade from a previous version must activate the Form Designer (com.glide.ui.ng. fd) plugin to use the feature

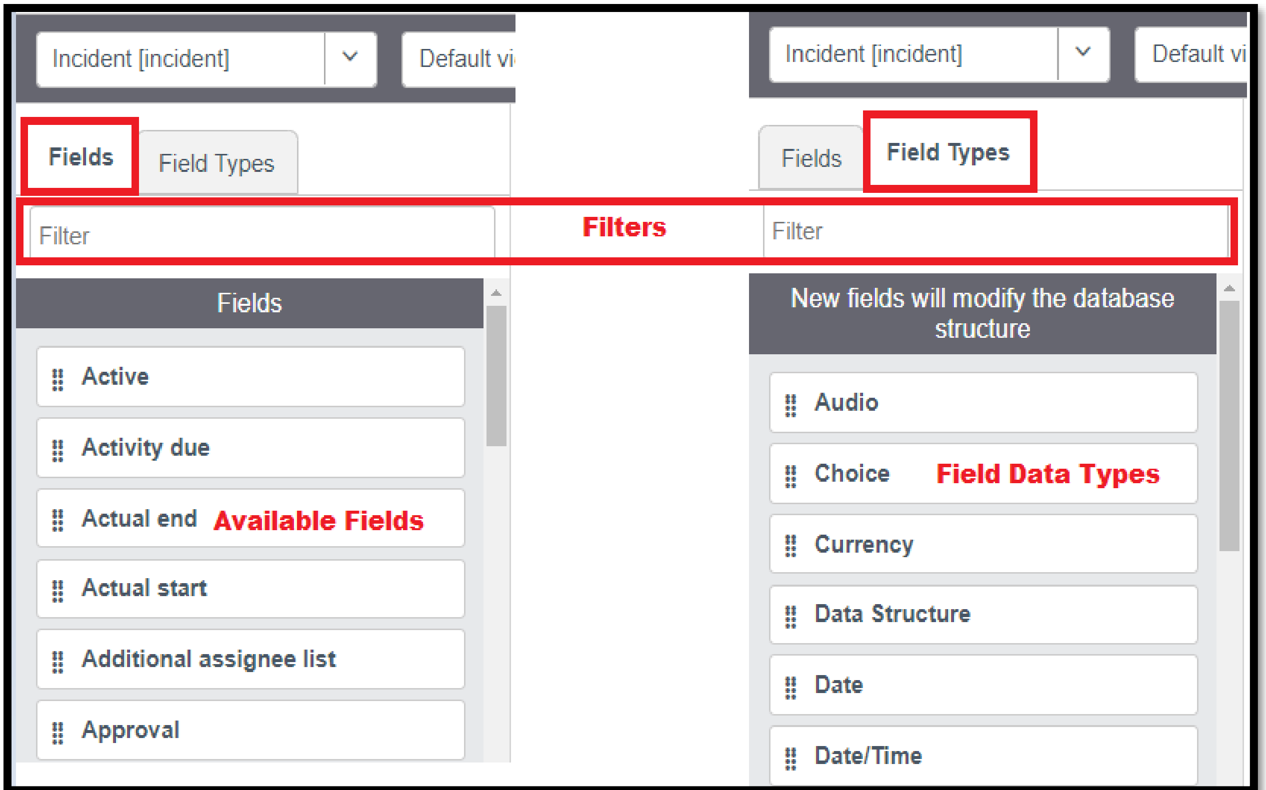
Parts of the UI form design interface



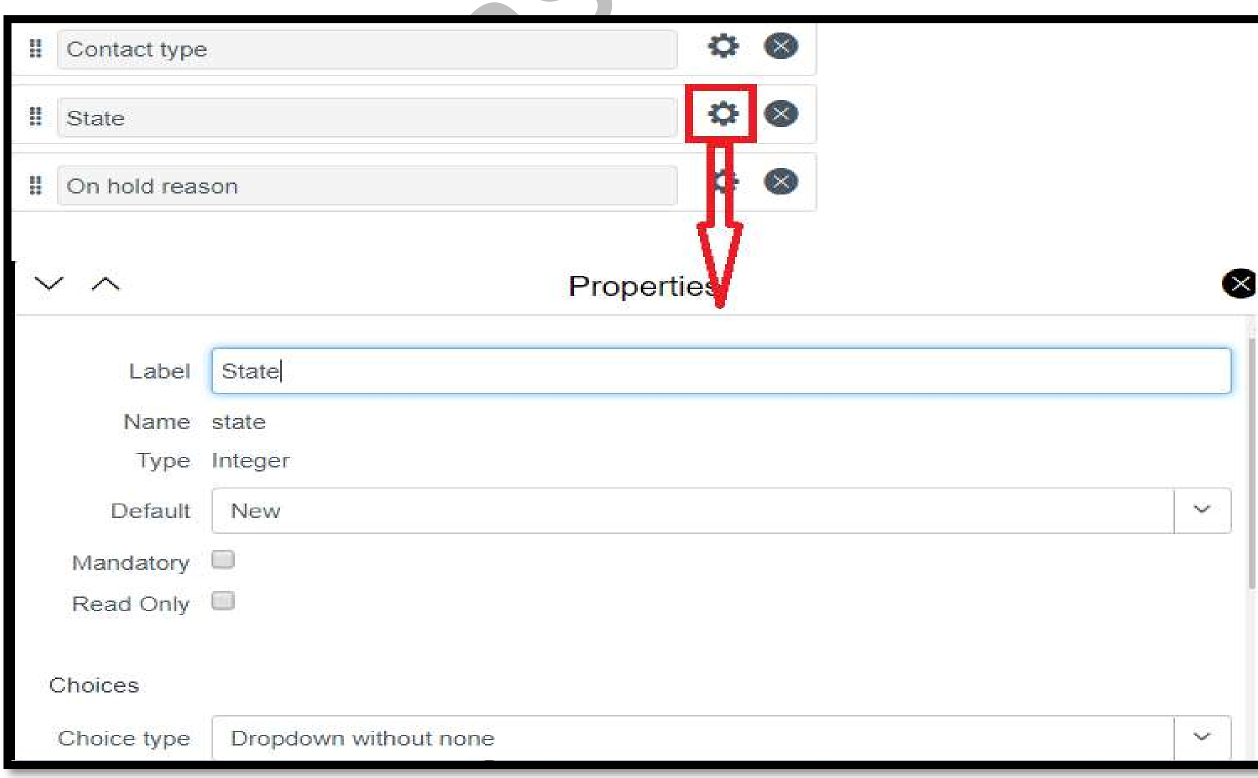
The field navigator

The field navigator allows you to manage fields on the form. It contains the following elements:

1. **Filter:** Allows text search on the currently selected tab.
2. **Fields:** Displays a list of existing fields you can add to the current form.
3. **Field Data Types**: Displays a list of field types you can add to the current form. Adding a field type to the form layout creates fields in the selected table when the form layout is saved.

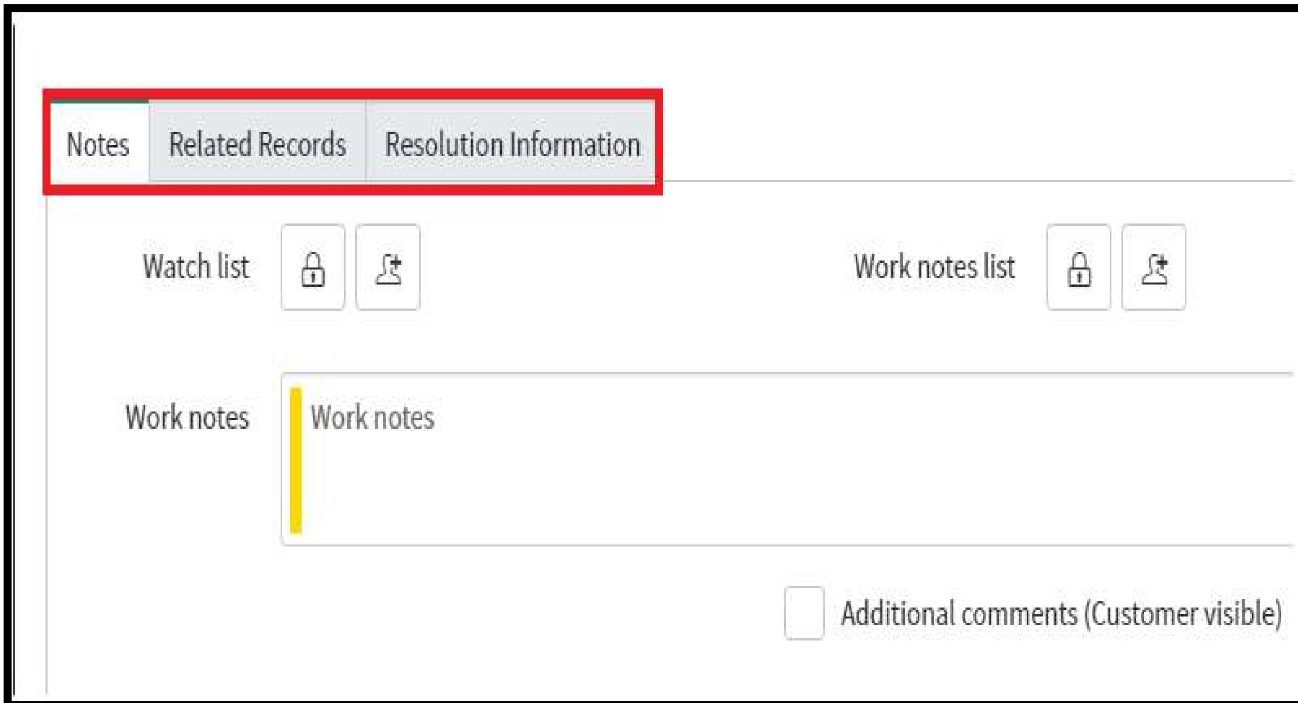


Field Properties



Working with Sections

Forms may contain one or more Sections. Sections are logical groupings of fields. In the baseline case, sections are rendered as tabs at the bottom of a form. The use of sections prevents users from having to scroll through long forms. Using a script, sections can be hidden when they are not needed.



Show or hide fields with the form designer

We can move fields to and from the form workspace to display the fields on the form.

Procedure

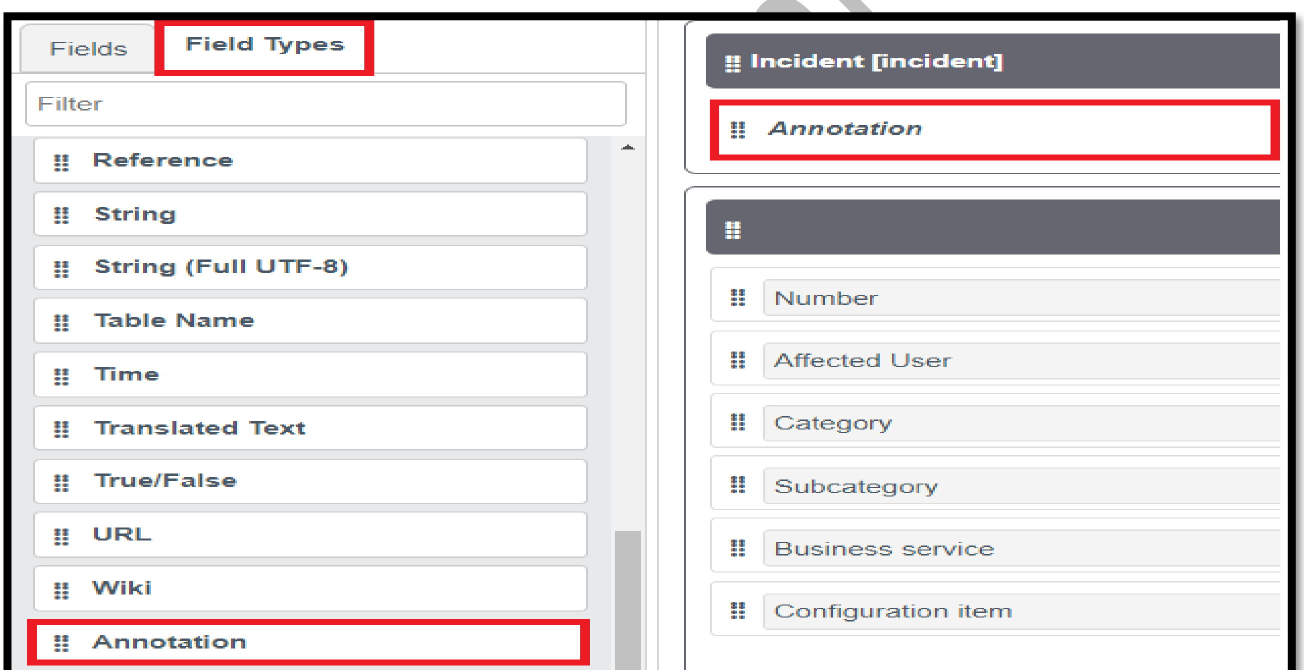
1. Navigate to the **form designer.**
2. From the list of fields in the Field Navigator, select the **Field** you want to display from the list of available fields.
3. Drag the field to the **form layout**.
4. Drag the field to the location you want it to be on the form.
5. To remove a field from the form layout, select the field and click the Remove this field **(X)** button beside the field label.
6. Click **Save.**

Exercise: 1 Add a form Annotation with the form designer

Annotations are form elements to add to display some data in visualization manner

Procedure

1. Navigate to the **Form Designer.**
2. Click the **Field Types** tab.
3. Select the **Annotations** field type.
4. Drag the annotation to the appropriate location on the form.
5. Point to the **Annotations field** and click the gear icon to open Properties dialog box.
6. In the Properties dialog box, enter a value for Annotation Text. You can use HTML tags to format the **Annotation text.**
7. Click **Save.**



Exercise :2 Working with New Section from form designer

Move sections to and from the form layout to display or remove the sections on the form.

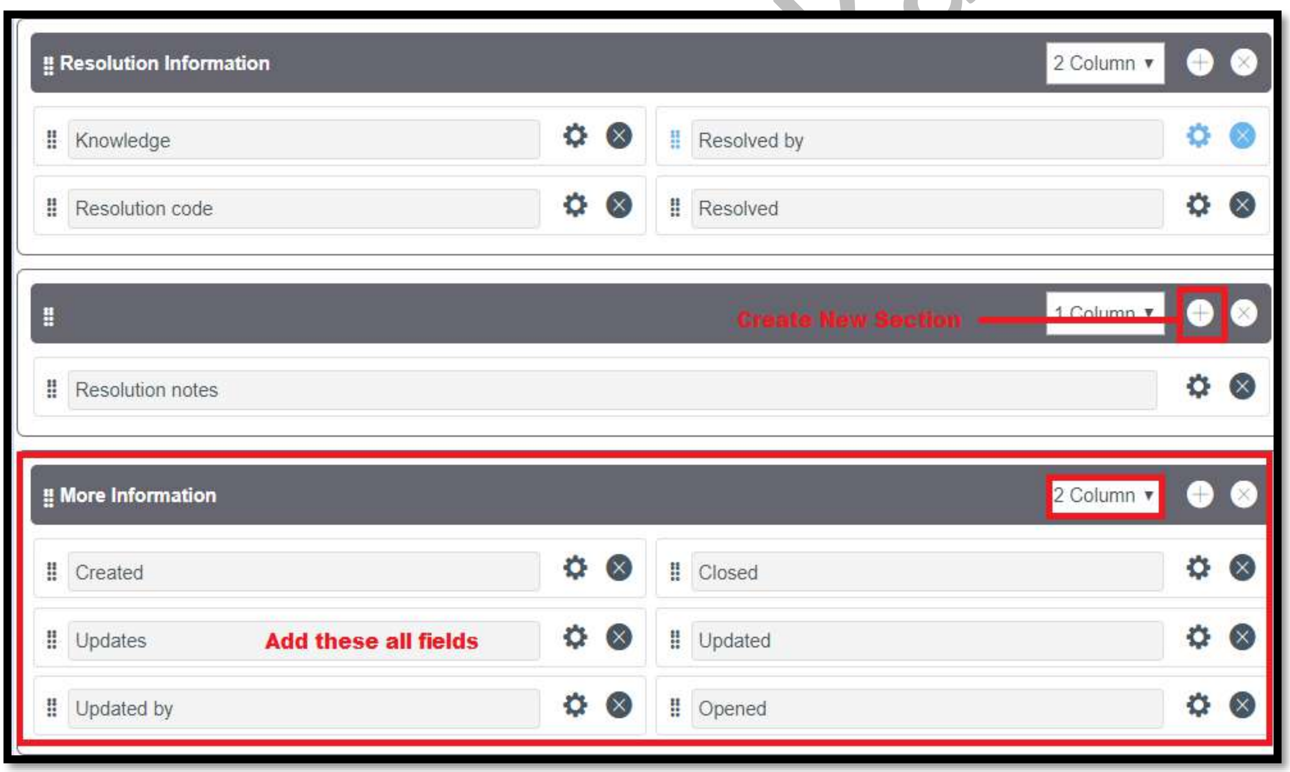
Note: Sections can have a **one- or two-column** layout.

By default, the first section on a form is always a read-only section displaying the label of the table. You cannot remove this section

Procedure

1. Navigate to the **form designer.**
2. Select an existing section.
3. Click the **(+)** button beside the section label to add a section.
4. Enter a label for the section.
5. **Drag sections** to reorder them. The label for the first section on the form designer is displayed as the form title.
6. To delete a section, click the **(x) button** beside the section label.
7. Click **Save.**

The following screen shot shows how you would use form designer to add and then delete a form section.



Working with formatters

Use formatters to add information from macros or previously scripted UI elements

What is Formatter

A formatter is a form element used to display information that is not a field in the record. Add formatters to a form by configuring the form.

Types of Formatters in Base Platform

Service now offering various type of formatters we can use for different requirement

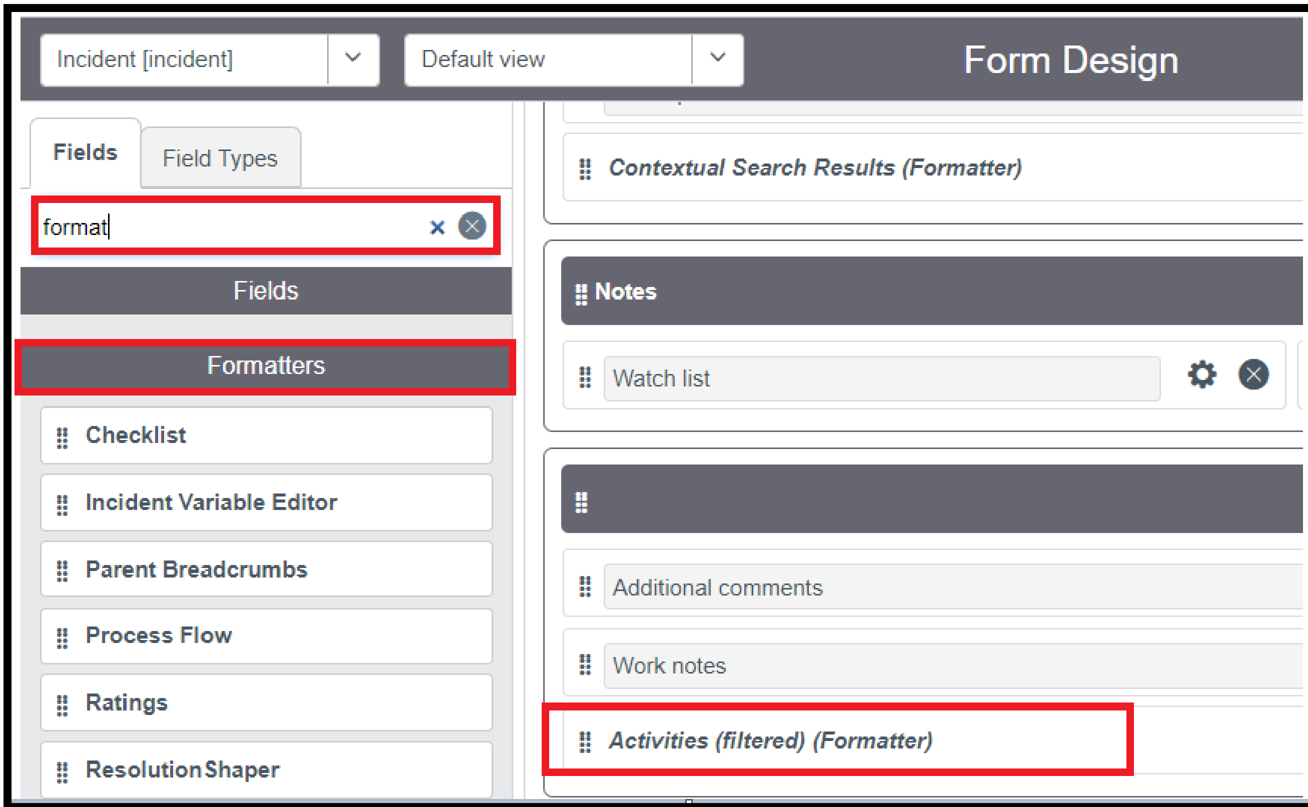
1. Activity Formatter
2. Process Flow Formatter
3. Parent breadcrumbs formatter
4. Approval summarizer formatter
5. CI relations formatter
6. More……

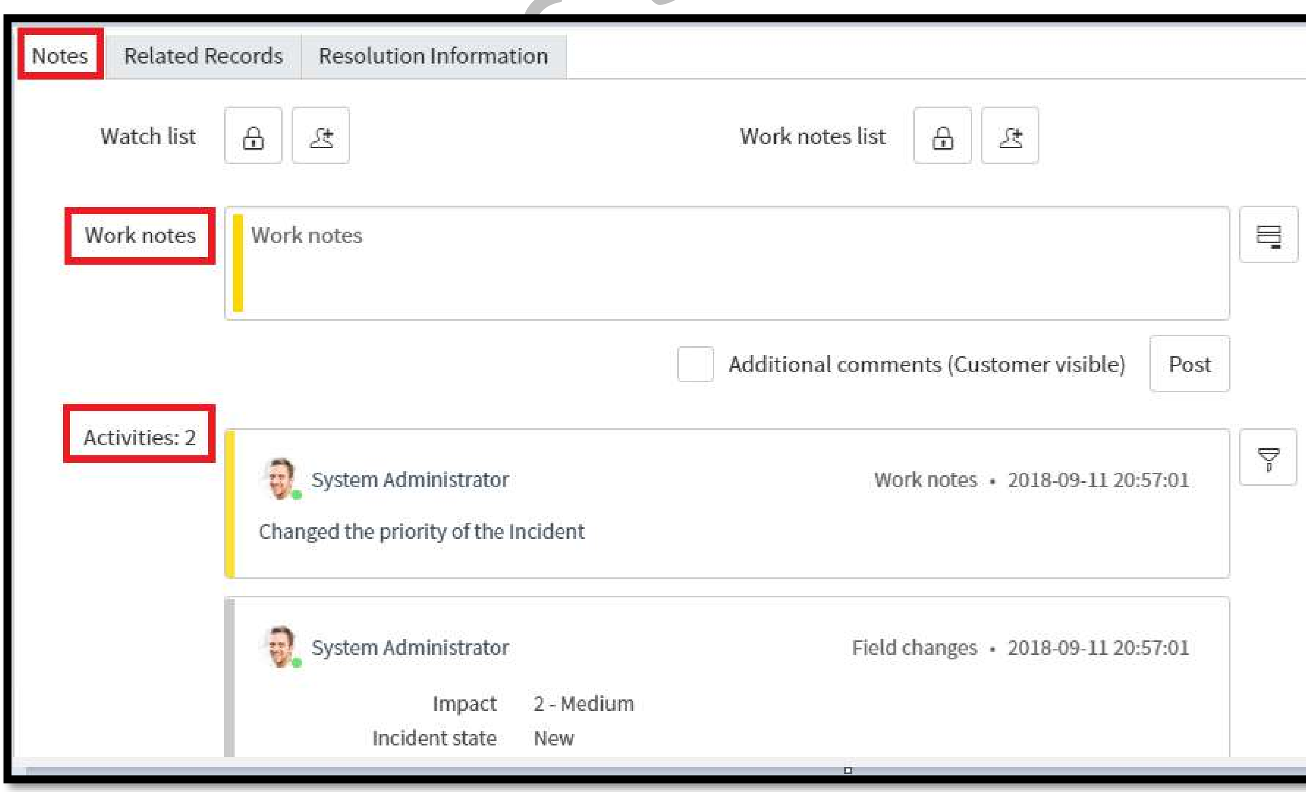
Exercise: 3 Configure Activity formatter

Displays the list of activities, or history, on a task form.

Procedure

1. Navigate to the **form designer.**
2. Click the **Fields** tab.
3. In the filter, enter the string **Format (Activity Formatter)**
4. Select the formatter you want to add to the form.
5. Add Activity Formatter in **Incident Table**
6. Drag the formatter to the location you want it to be on the form.
7. Click **Save.**


Incident [incident]
Default view
Form Design
Fields
Field Types
format
Fields
Formatters
Checklist
Incident Variable Editor
Parent Breadcrumbs
Process Flow
Ratings
Resolution Shaper
Contextual Search Results (Formatter)
Notes
Watch list
Additional comments
Work notes
Activities (filtered) (Formatter)

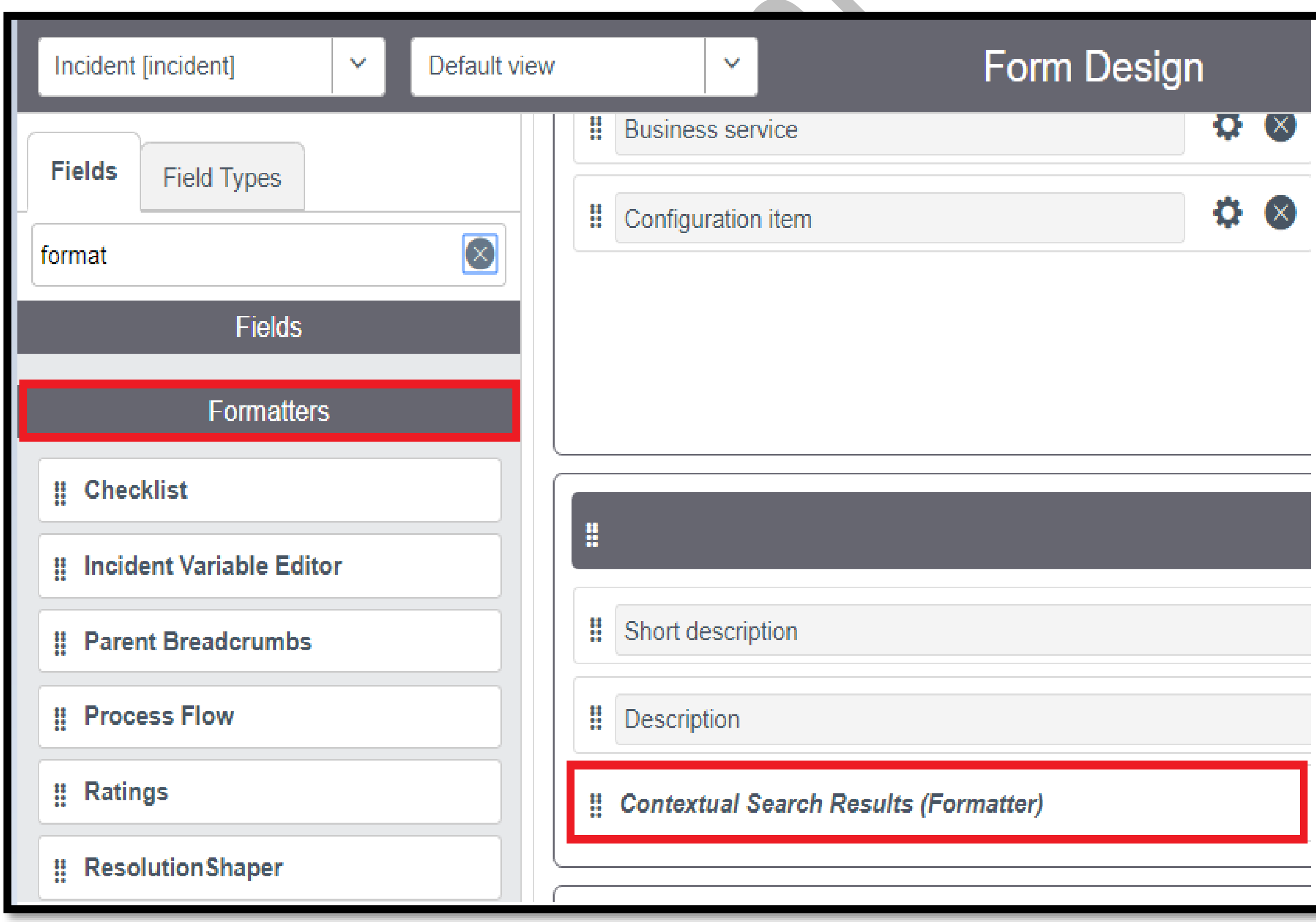

Notes
Related Records
Resolution Information
Watch list
Work notes list
Work notes
Work notes
Additional comments (Customer visible)
Post
Activities: 2
System Administrator
Work notes • 2018-09-11 20:57:01
Changed the priority of the Incident
System Administrator
Field changes • 2018-09-11 20:57:01
Impact 2 - Medium
Incident state New

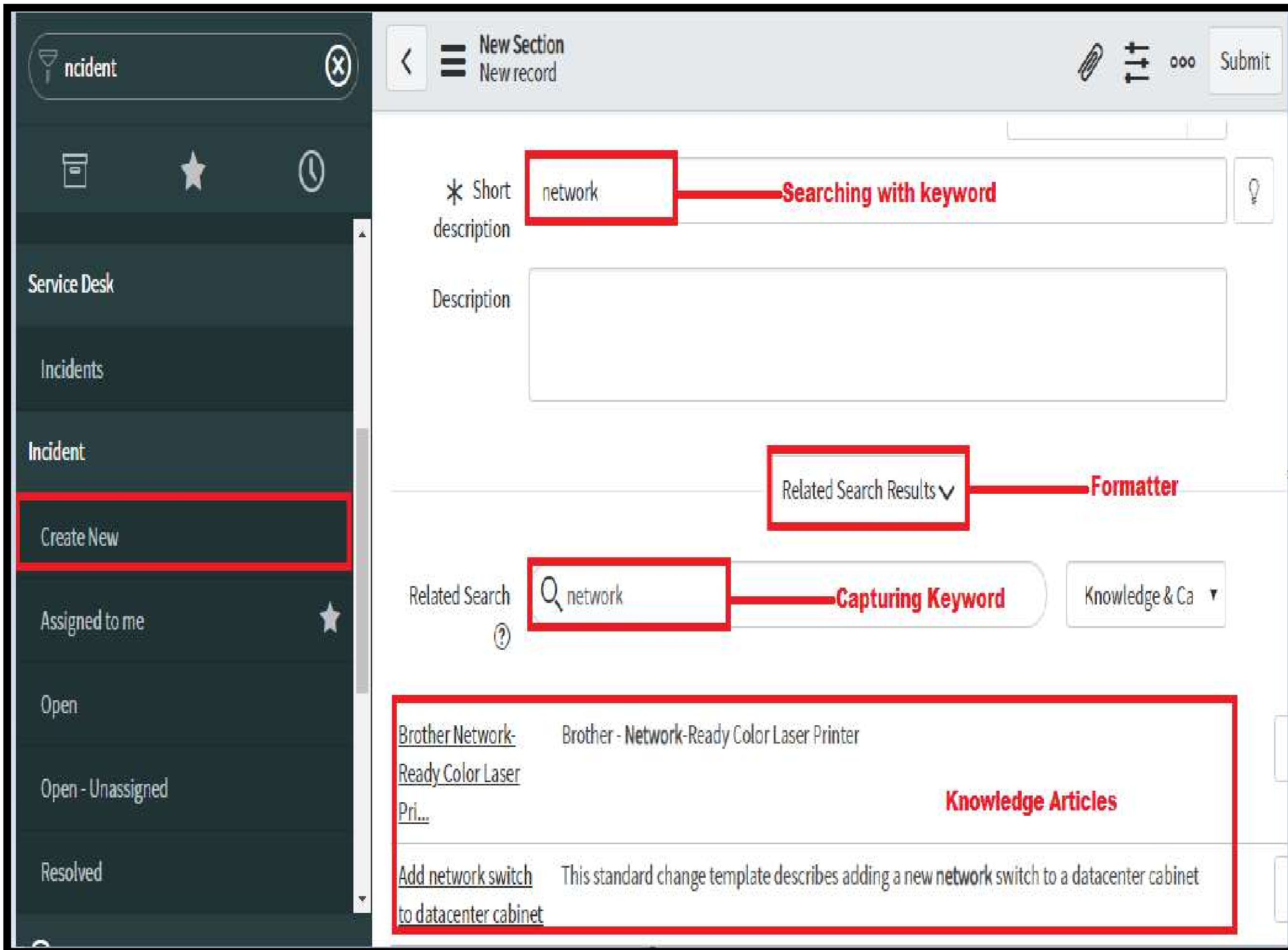
Exercise: 4 Contextual Search Result(Formatter)

The contextual search results formatter can apply for search criteria and **knowledge articles** and **service catalog items**

Procedure

1. Navigate to the **form designer.**
2. Click the **Fields** tab.
3. In the filter, enter the string **Format (Contextual Search Result)**
4. Select the formatter you want to add to the form.
5. Add Activity Formatter in **Incident Table**
6. Drag the formatter to the location you want it to be on the form.
7. Click **Save.**

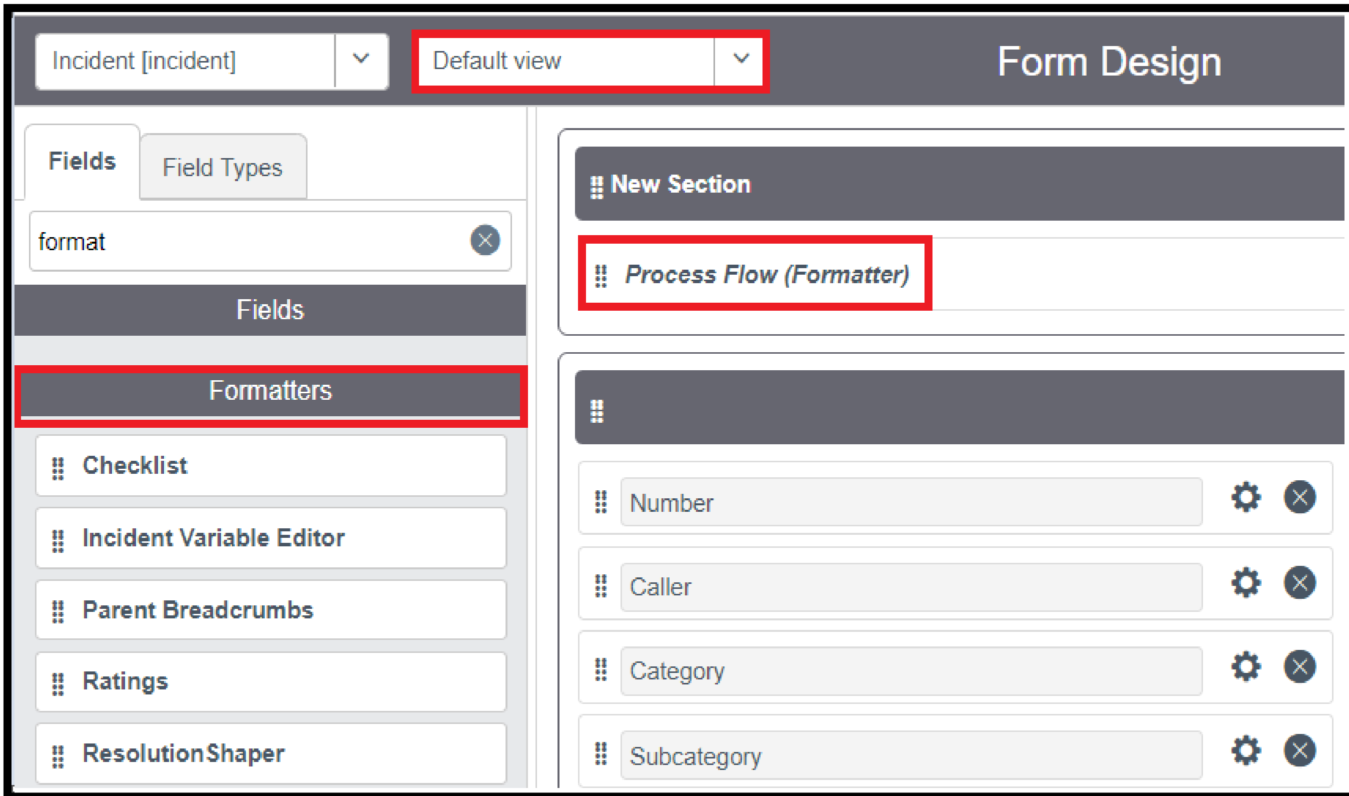




Exercise: 5 Process Flow (Formatter)

Procedure

1. Navigate to the **form designer.**
2. Click the **Fields** tab.
3. In the filter, enter the string **Format (Process Flow)**
4. Select the **formatter** you want to add to the form.
5. Add Activity Formatter in **Incident Table**
6. Drag the formatter to the location you want it to be on the form.
7. Click **Save.**

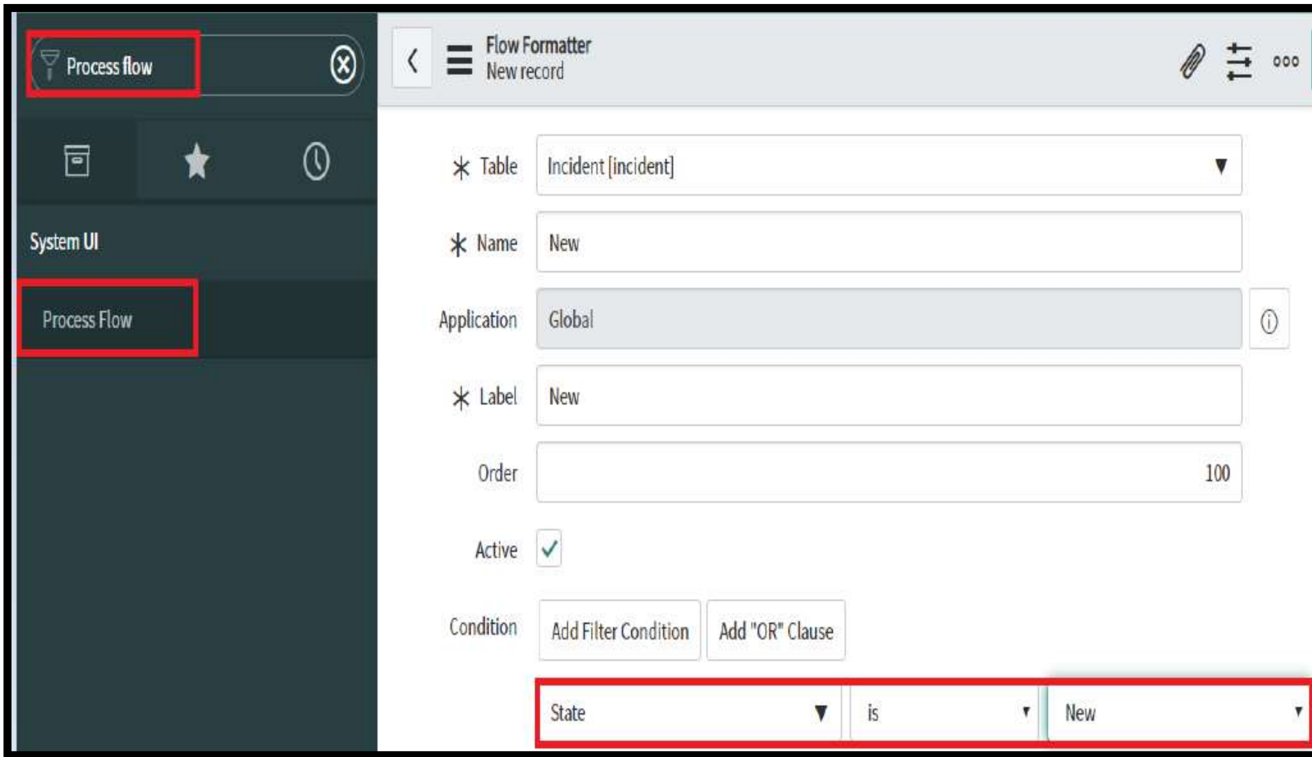


Note: In case if **Process Flow formatter** is not available in base instance then we need to Create here

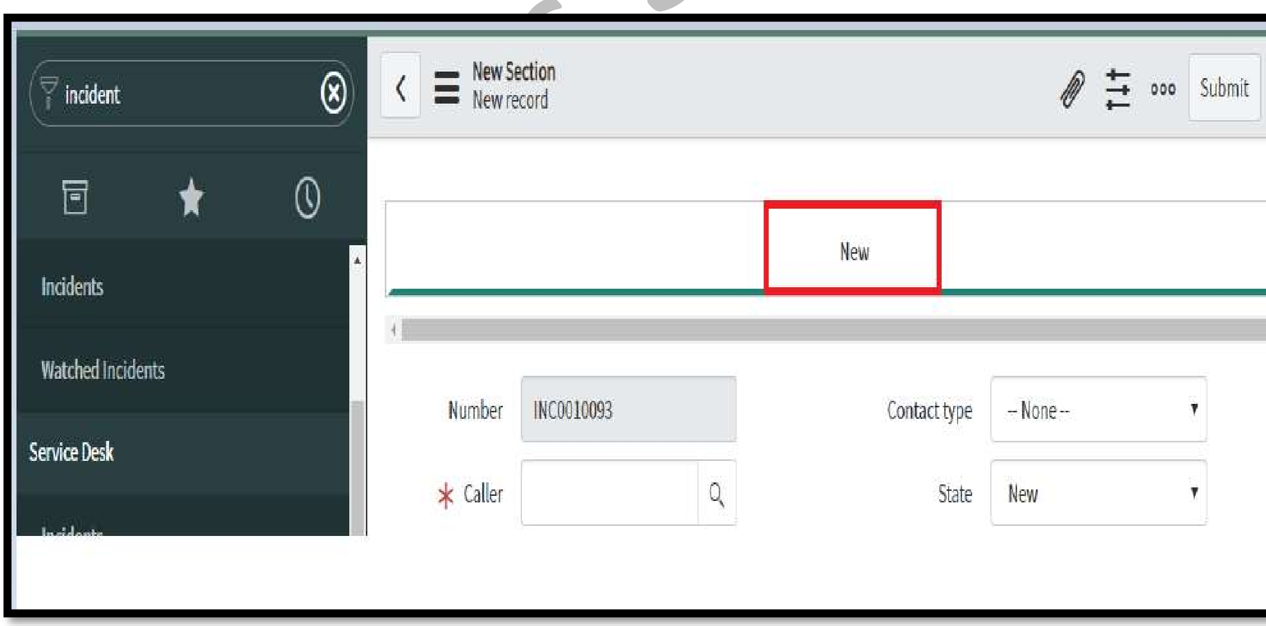
Exercise: 6 Process flow Formatter Configure in Incident Table

Procedure

1. Navigate to the **SystemUI→Process Flow**
2. Fill the form table name (**Incident**)
3. Name: **New**
4. Label Name: **New**
5. Order Number: **100**
6. Condition: **[State] [is] [New]**
7. Drag the formatter to the location you want it to be on the form.
8. Click **Submit**

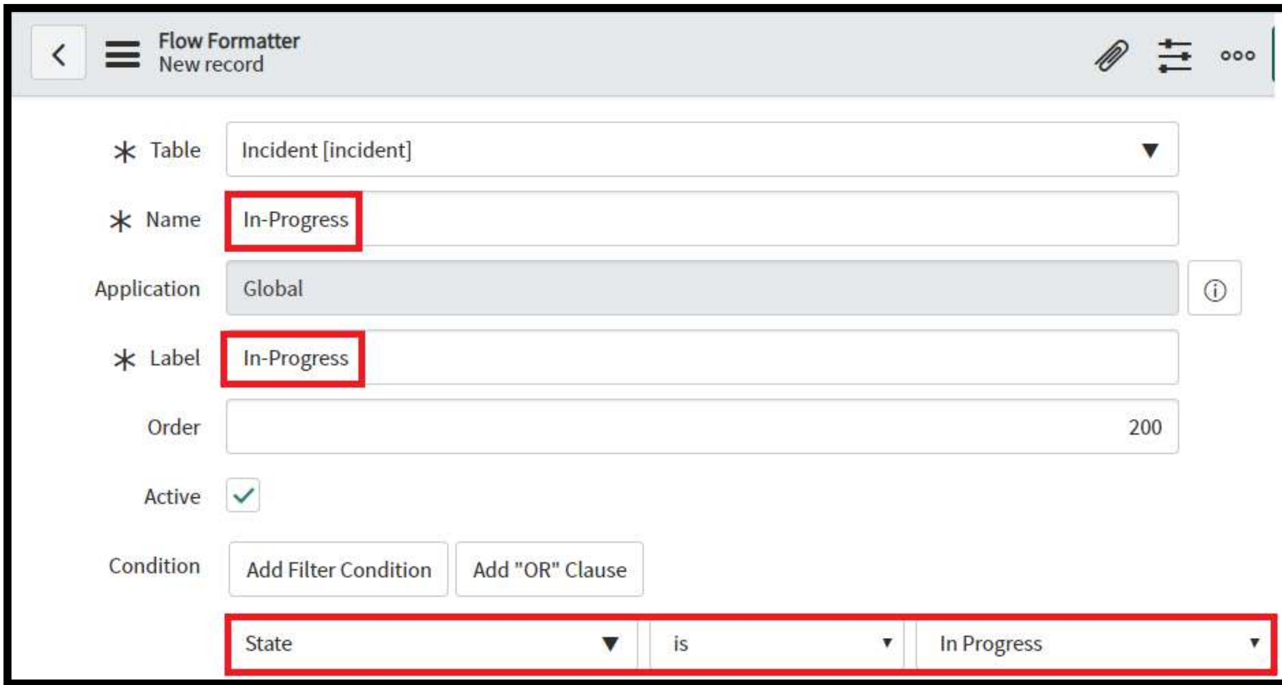


Navigate to **Incident** view check that added formatter

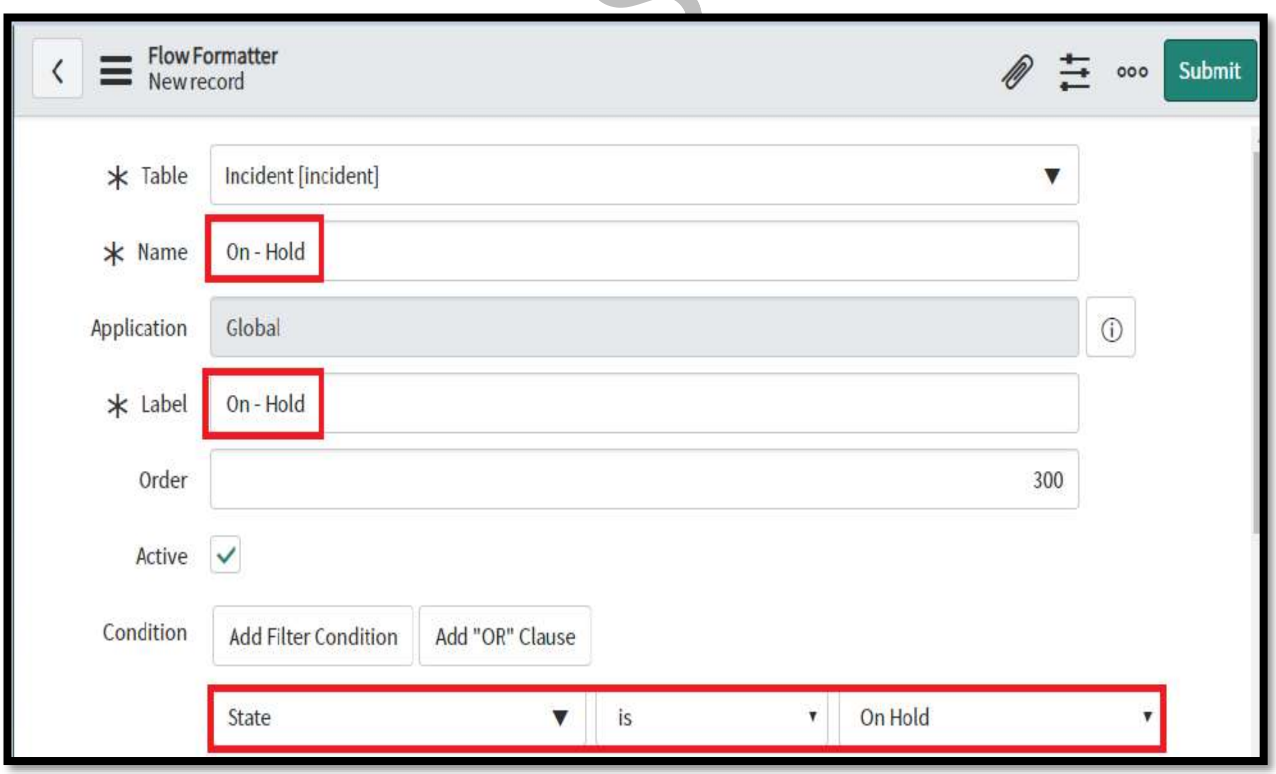


Similar to add reaming status

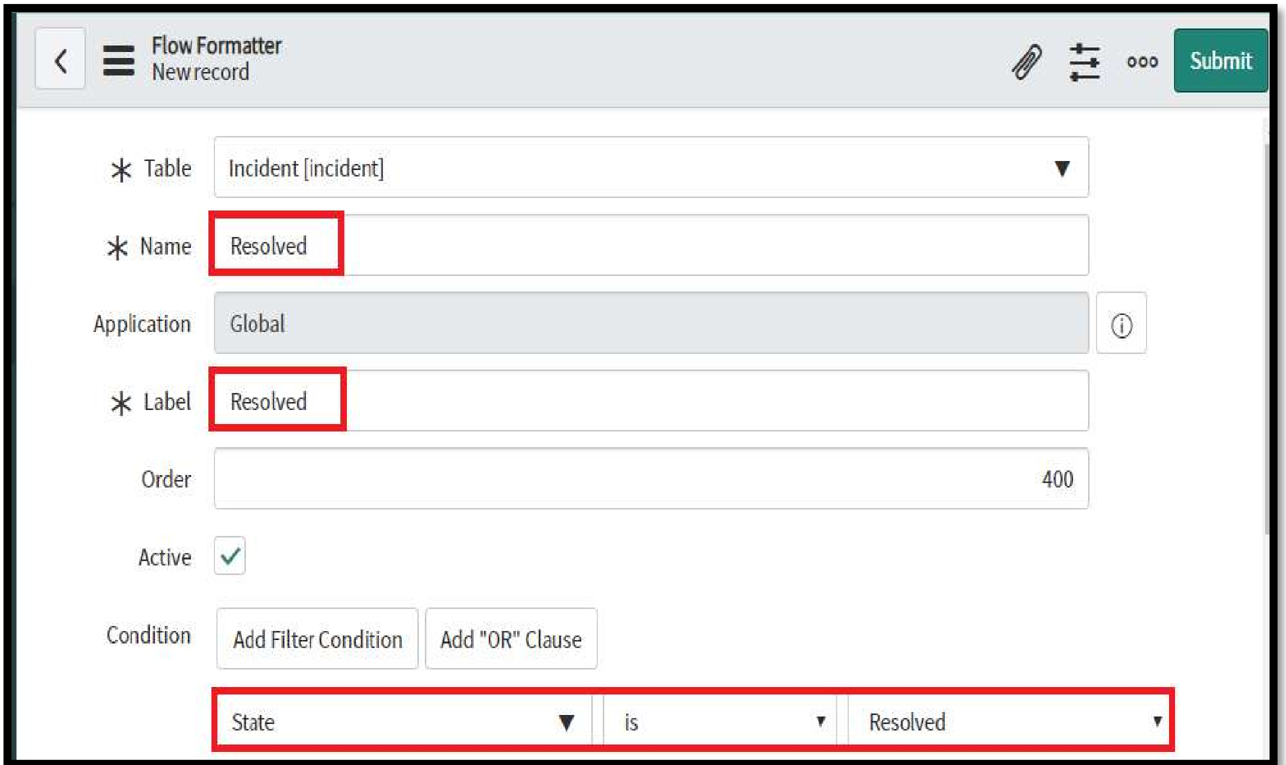
1. **[State] [Is] [In-Progress]**



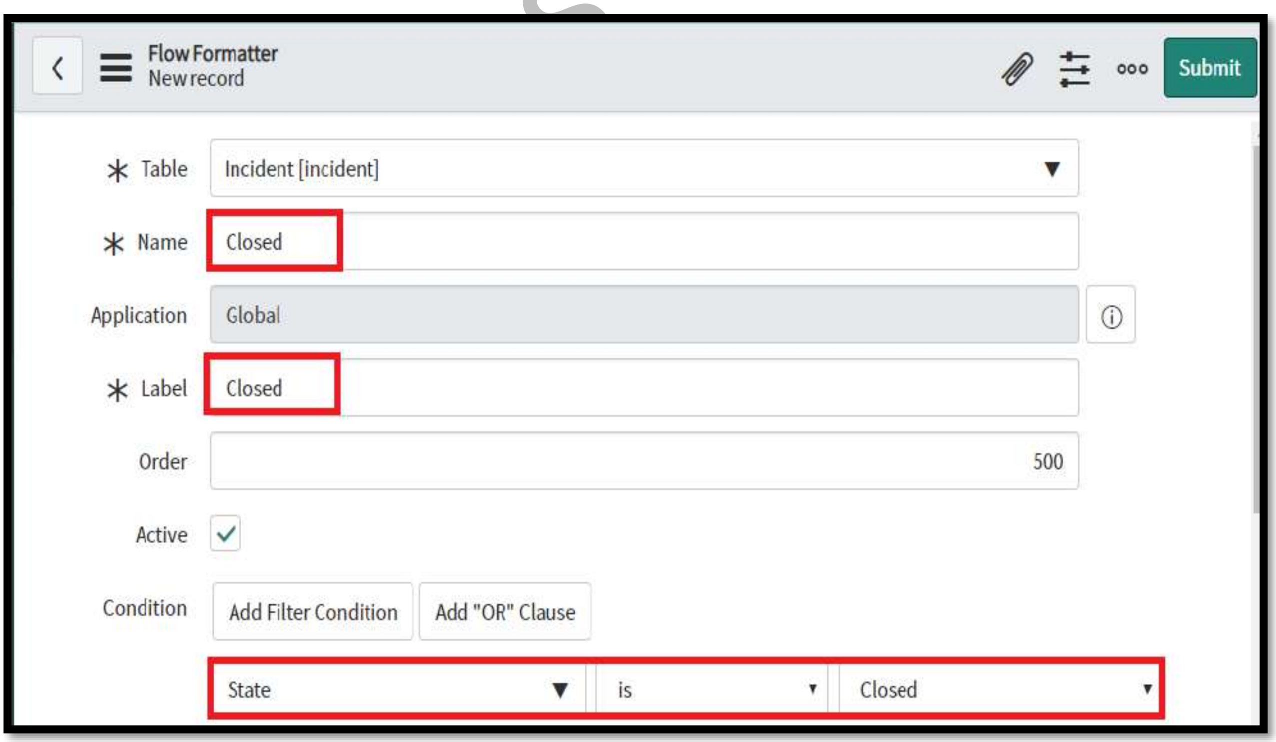
2. **[State] [Is] [On-Hold]**



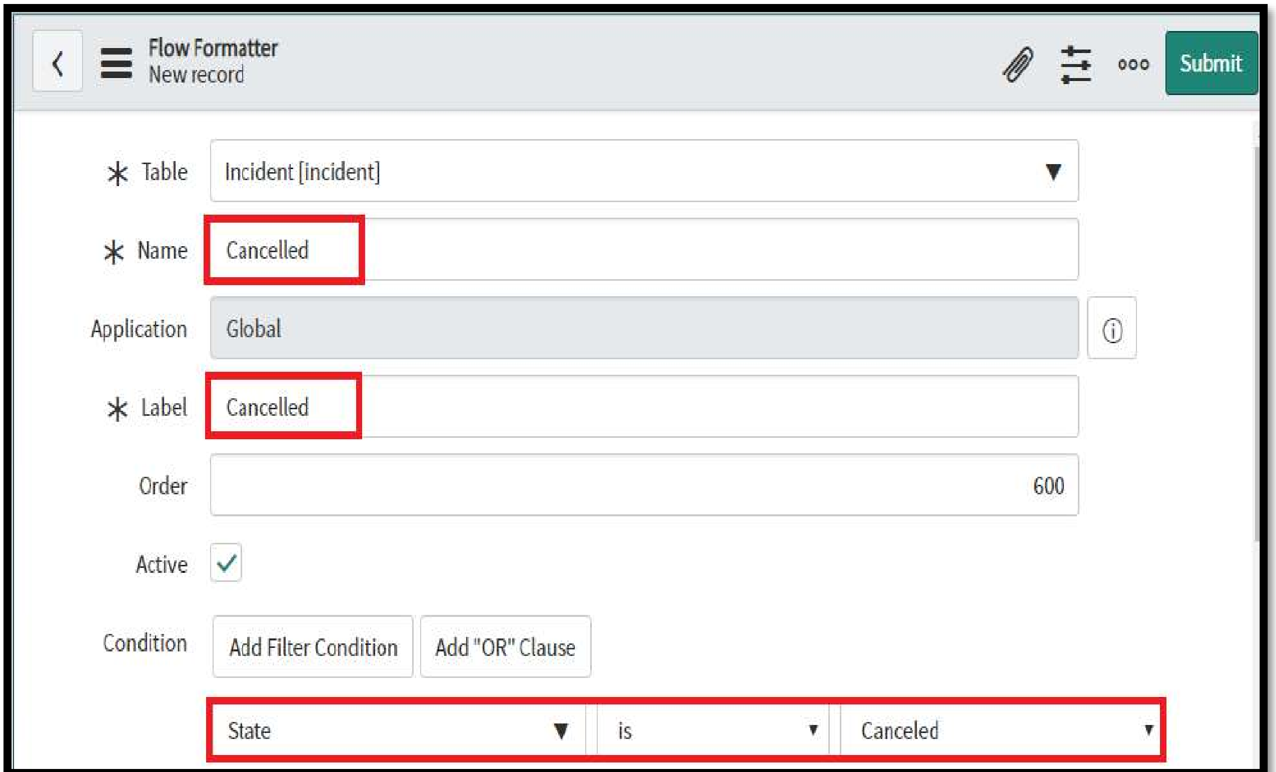
3. **[State] [Is] [Resolved]**

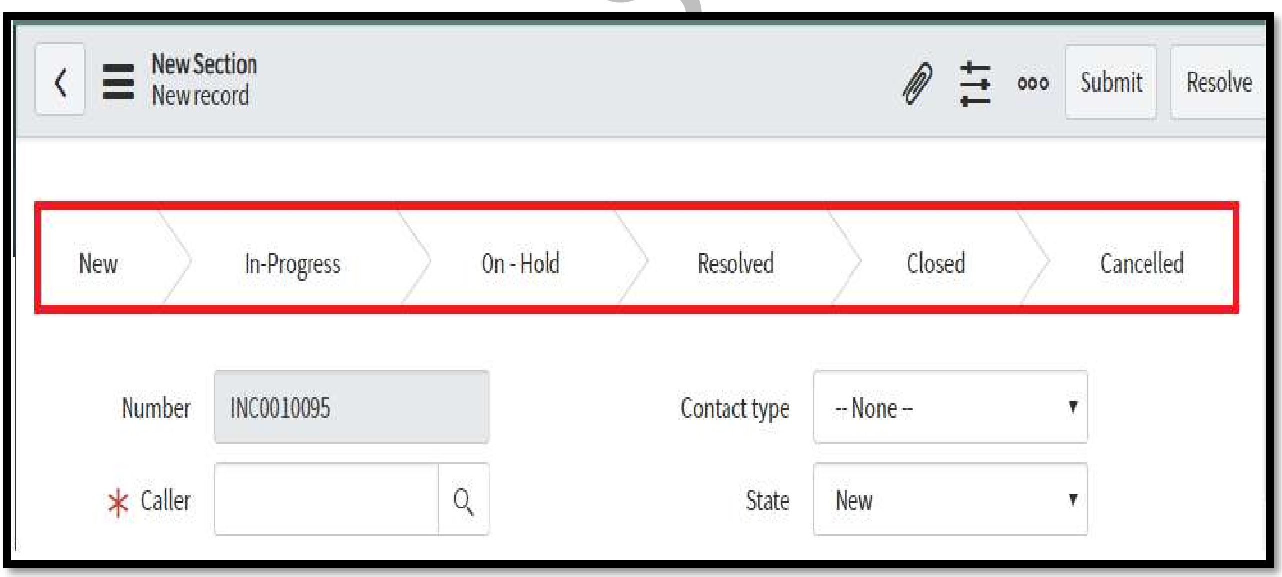


4. [State] [Is] [Closed]



5. [State] [Is] [Cancelled]

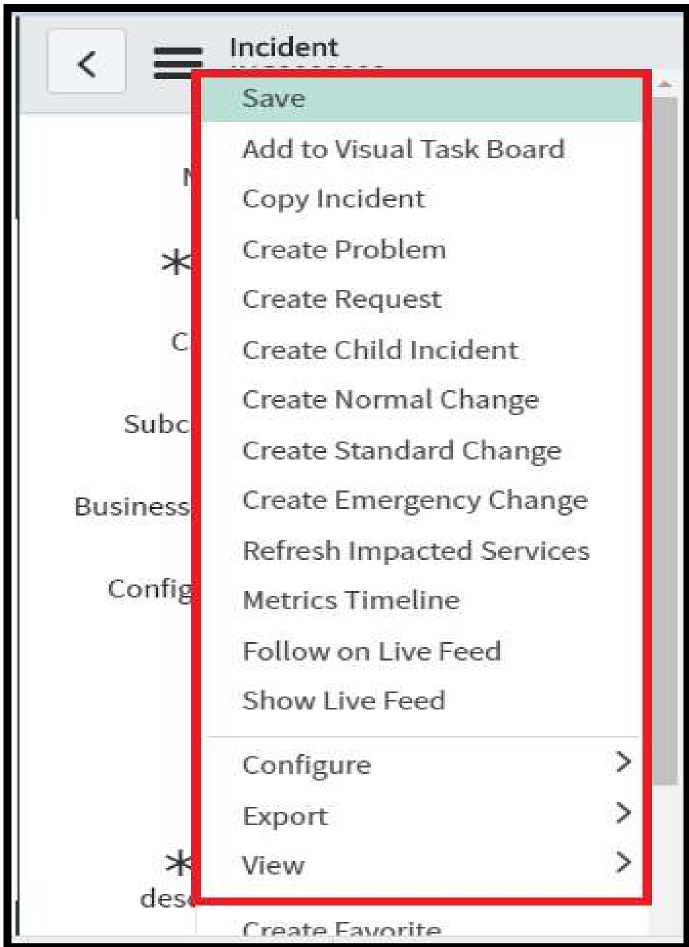




Form Context Menu

The form context menu provides controls based on the table and user access rights. Administrators can customize some of the options available on a context menu using UI actions.

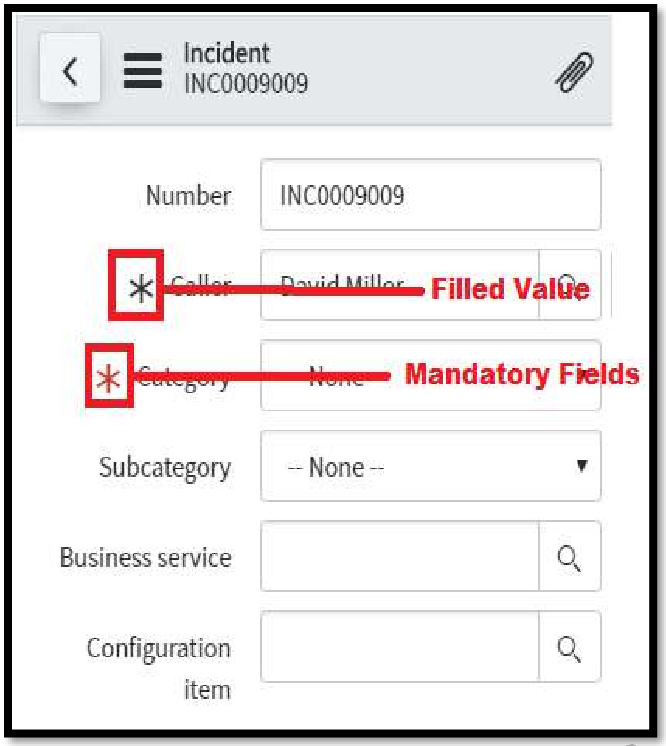
Access the form context menu by clicking an icon beside the form title or by right-clicking the form header



Field status indicators

Indicators are used on some fields to denote a special field type.

A field status indicator is a colored asterisk that may appear to the left of mandatory fields. Field status indicators change colors to represent different states of mandatory fields.



Working with Dot-walking in List

We can access fields on a related table from a **form, list, or script** by dot-walking.

We can dot-walk to related fields in a list, such as the field list in a filter. This example can define how to filter the Incident **[incident]** table by the company of the caller who registered the incident.

List fields Dot. Walking

We can dot-walk to related fields in a **list**, such as the field list in a filter. This example can explain how to filter the Incident **[incident]** table by the company of the caller who registered the incident.

When we open the list of fields that we want to filter, we see the list of available Incident table fields. The reference fields are followed by the related fields.

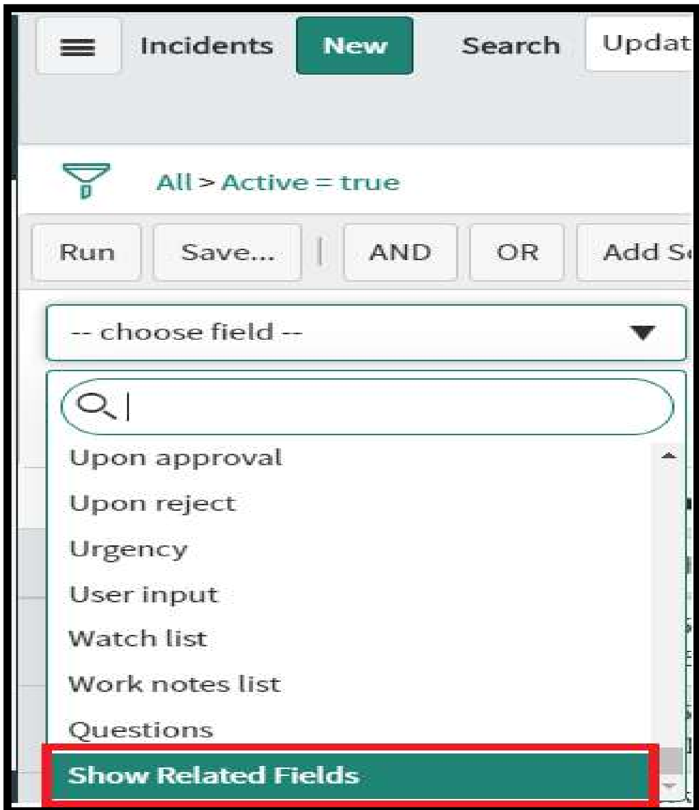
For example, **Caller** is followed by **Caller > User fields**, which means that **Caller is a reference field,** and the related fields are user fields on the **Caller record**. If the related fields are not present in the list, you would select Show Related Fields at the bottom of the list.

Exercise: 7 Configure Dot-walking in Lists

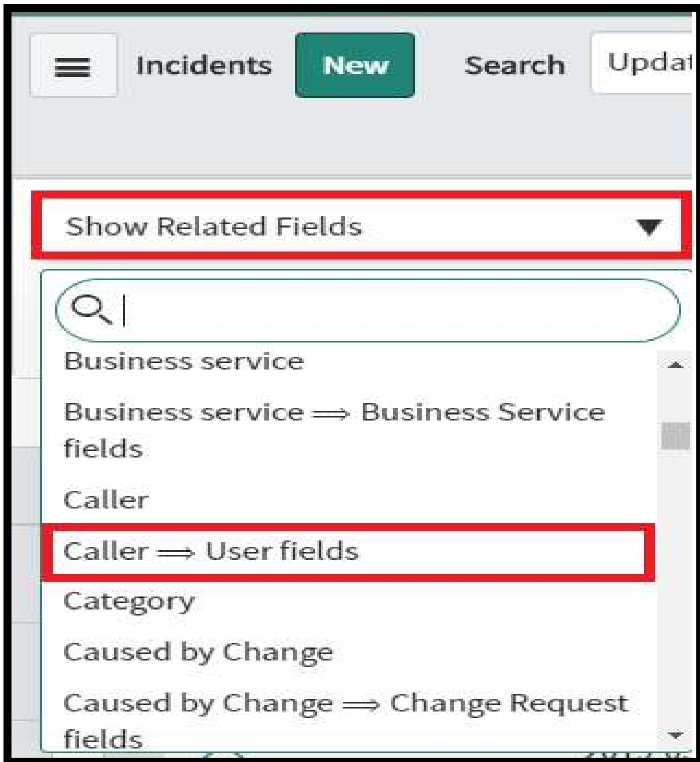
We can configure **dot. walking** for List apply to apply filter condition for referenced fields

Procedure

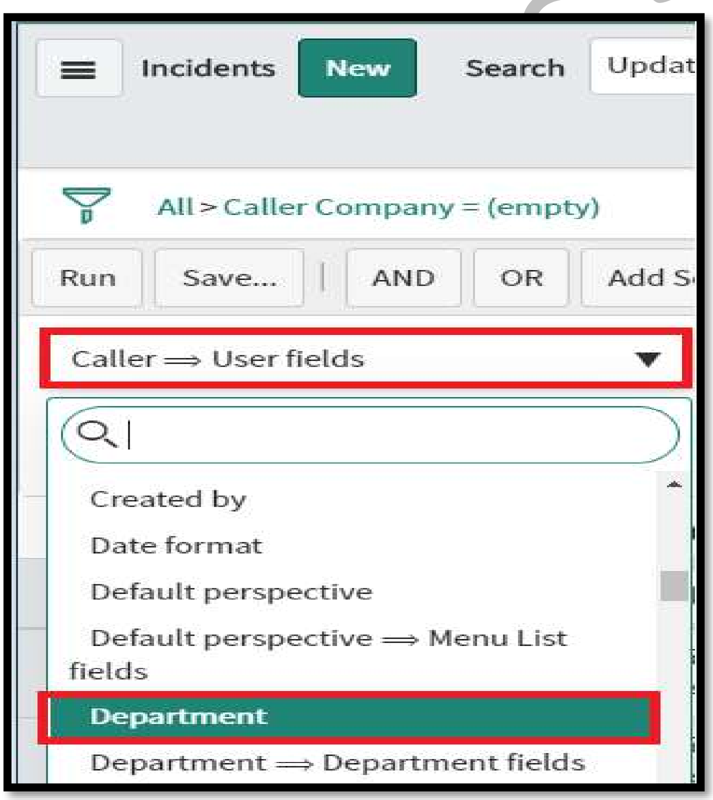
1. Navigate to **Incident list > Open Condition Builder**
2. Select **Show Related Fields**



3. When we select **Show Related Fields**, the menu can reload the list to display related fields.

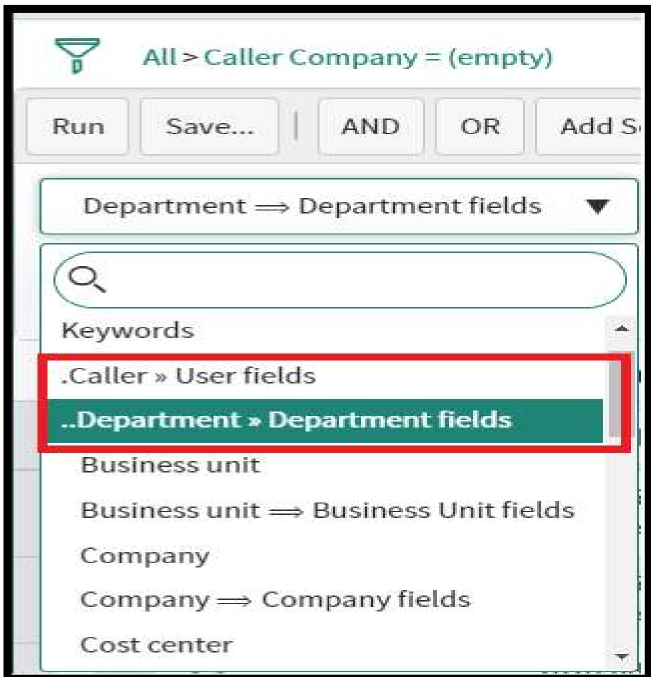


4. Select **Caller ==> User fields**



5. When we selected **Department** under **Caller → User fields**, the field then becomes **Caller.Deparment.**

The following example shows where you are in the **dot-walk.** Each selected reference is stored at the top of the fields menu, and the number of dots preceding the field label indicate how many dots from the initial record the user has reached.



This example can show that the user is at **Incident. Caller. Department**. We can return to higher levels in the hierarchy by selecting fields located at the top of the menu. For instance, selecting **Incident fields** returns to the list of incident fields.

The related fields can be removed by selecting **Remove Related Fields** at the bottom of the list.

Exercise: 8 Configure Dot-walking in Forms

The tree picker interface presents an expandable, hierarchical view that may be used when selecting fields. **Dot-walking** help to access to fields on **Related Tables** from a form, list, or script.

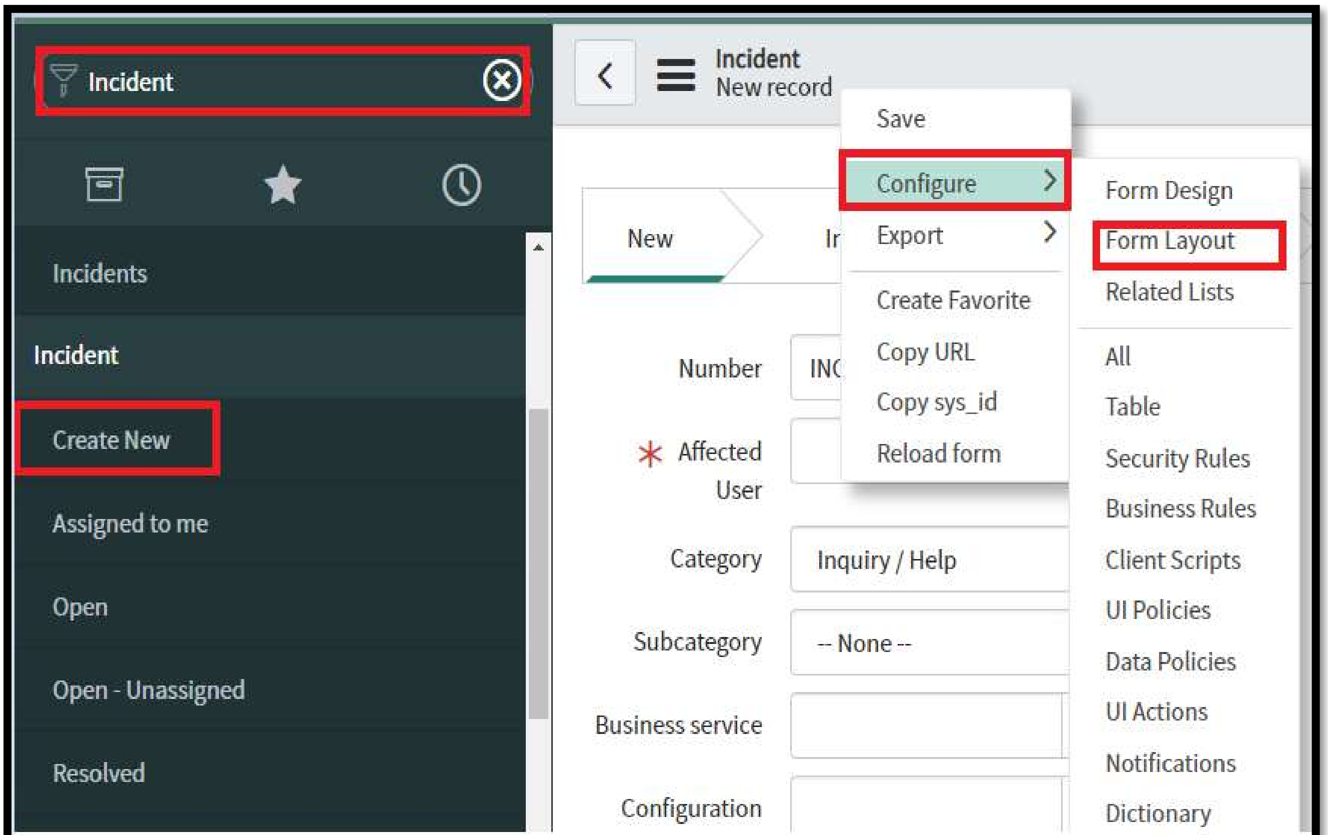
If the current table contains any reference to another table, any field on the referenced table can be accessed using **dot-walking.**

Dot-walking can reference a field by building a chain model of field names separated by **dots** (periods). For instance, **incident. caller. department.** references the company of the user caller to an incident.

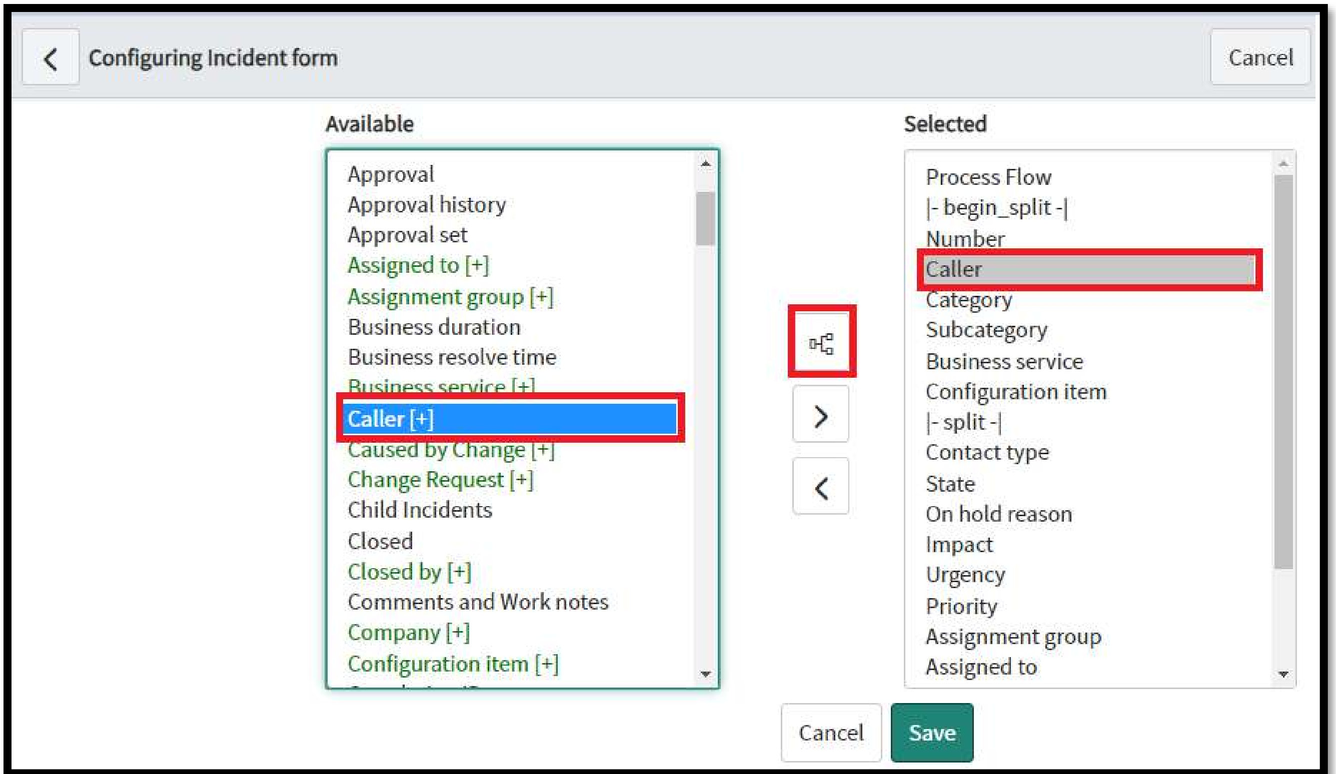
Note: The recommended limit for chain length is **Three Levels.**

Procedure

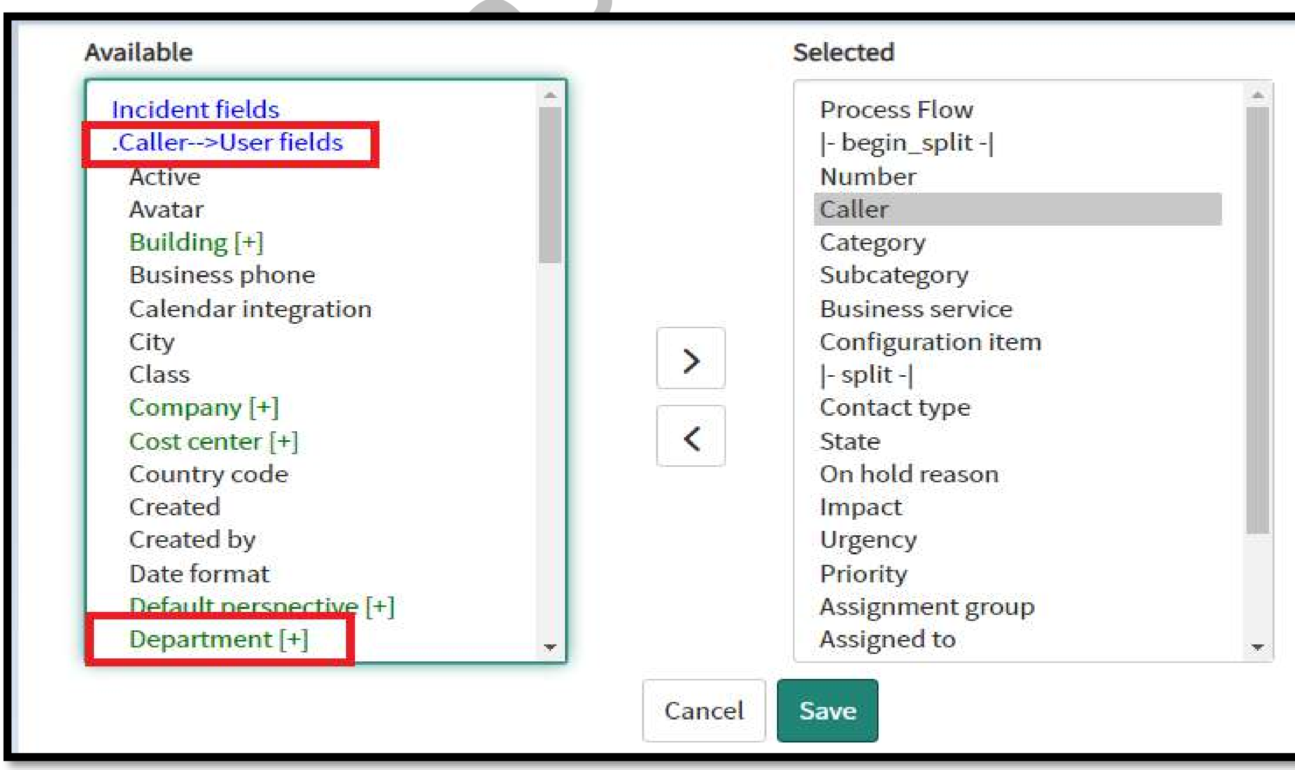
1. Navigate to **Incident Record**
2. Click on **Create New**
3. Right click on **Form Header**
4. **Configure > Form Layout**



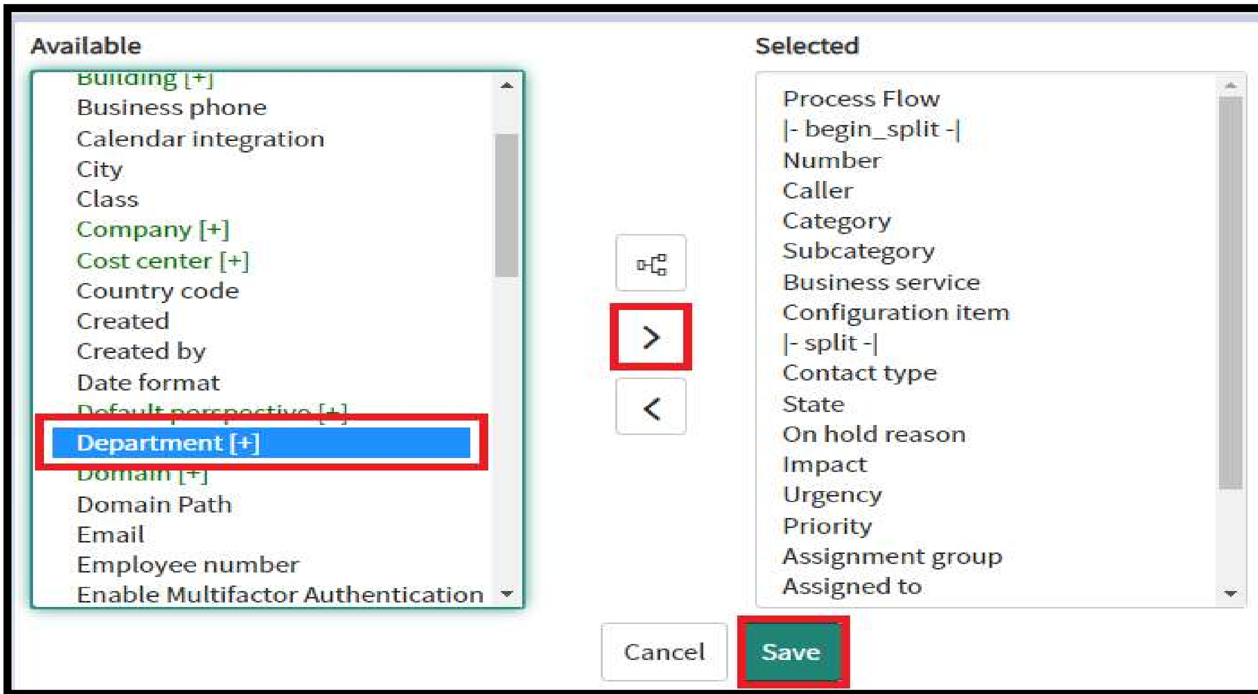
5. Select **Caller** field
6. Click on **Expand Reference field**

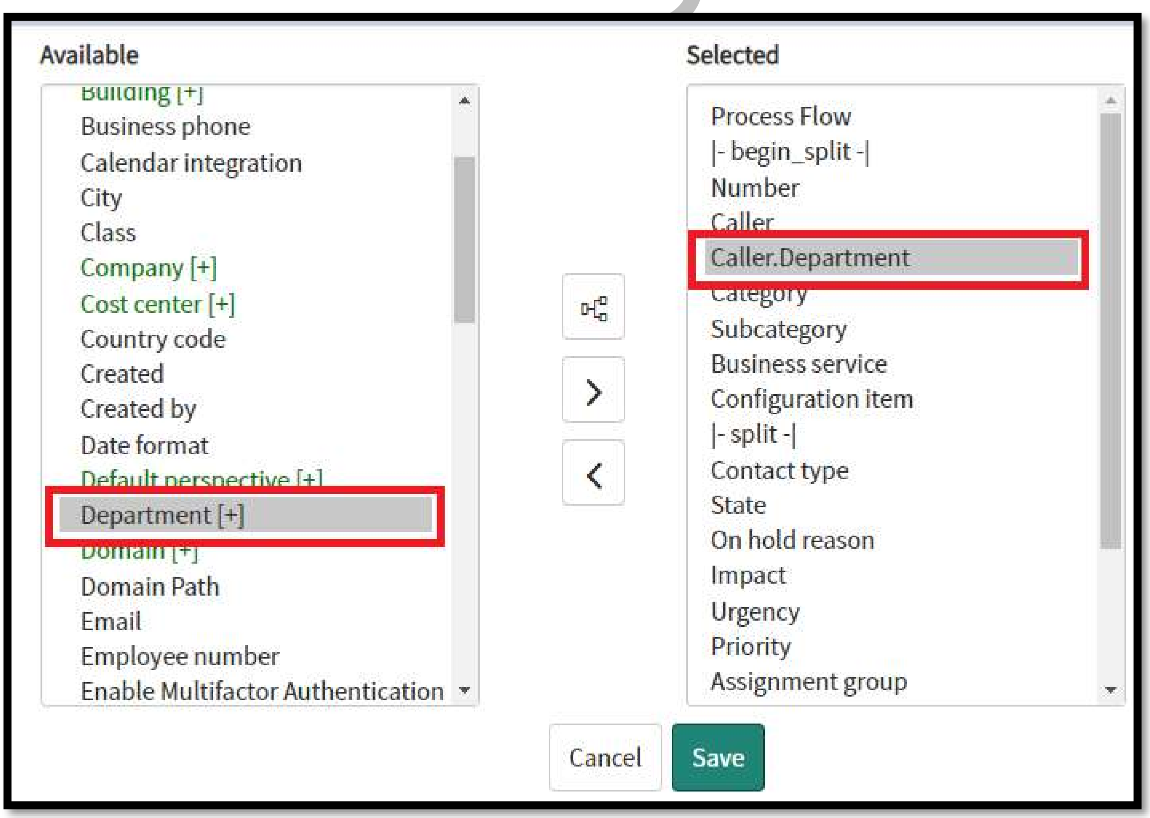


7. Open **.Caller→User fields**
8. Select **Department** field

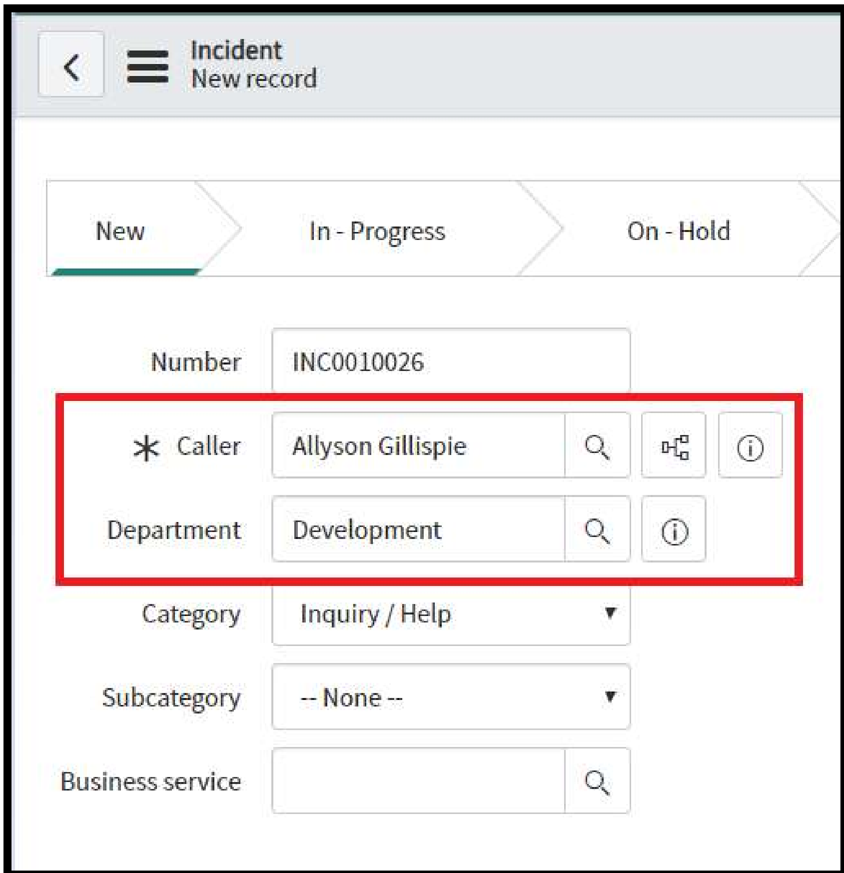


9. Click on **Add**
10. **Save**





11. Navigate to **Incident Form**
12. Select any **Caller Value** from caller field
13. Automatically populated **Department Value**



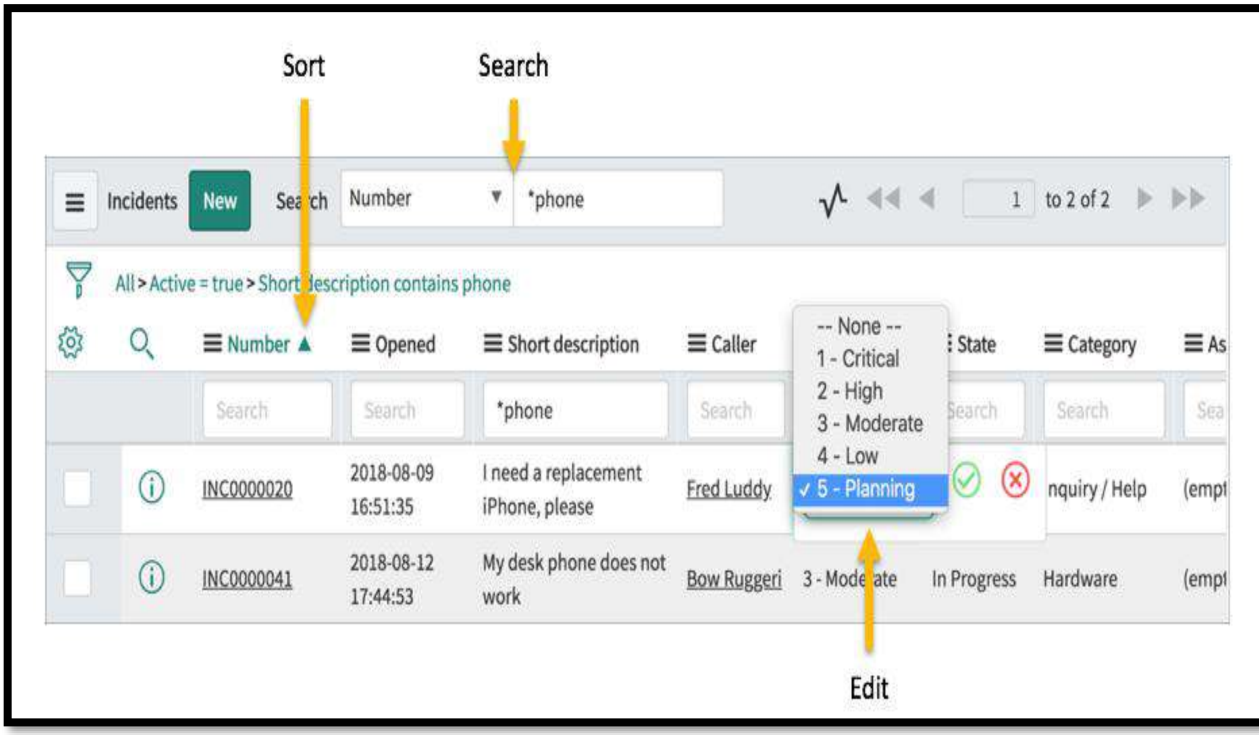
Chapter – 5

Lists

What is List ?

A list is a content page displaying zero or more records from the same table. Rows and columns organize the list. Each row is a record and each column is a field from the record. Lists are:

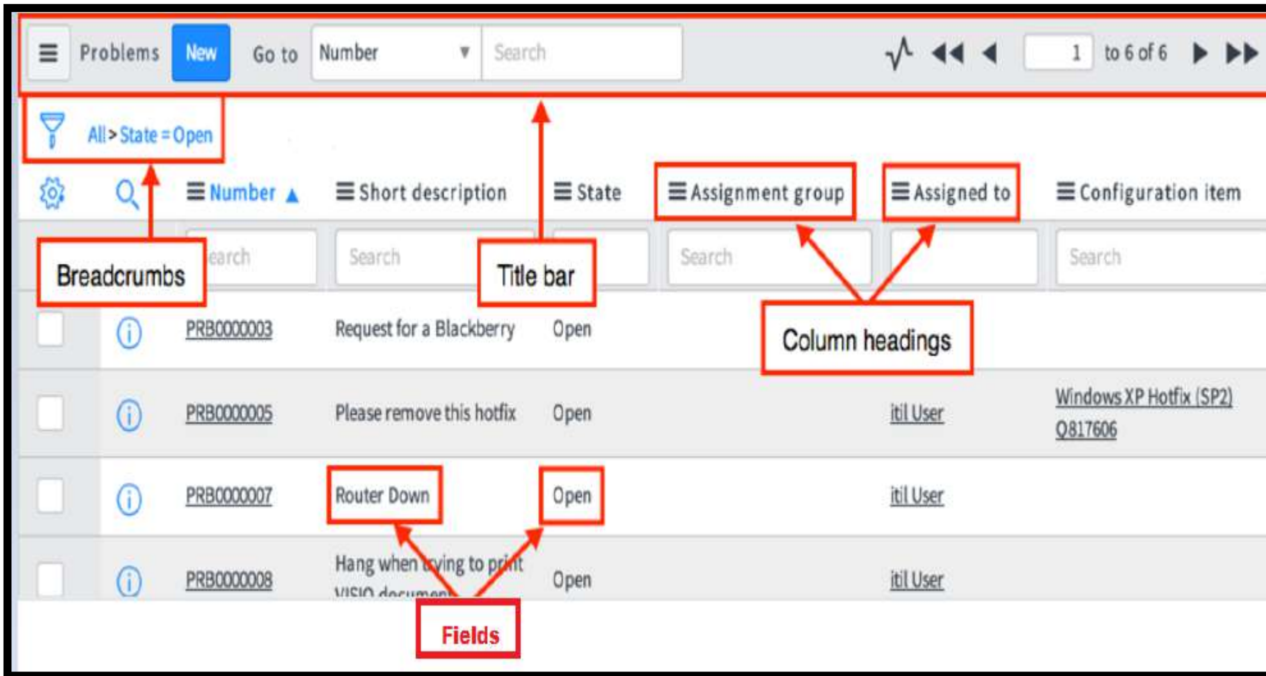
1. **Searchable:** Enter a value in the Search field
2. **Sortable:** Click the column label
3. **Editable** (if permissions allow): Double-click a field value



A list displays a set of records from a table.

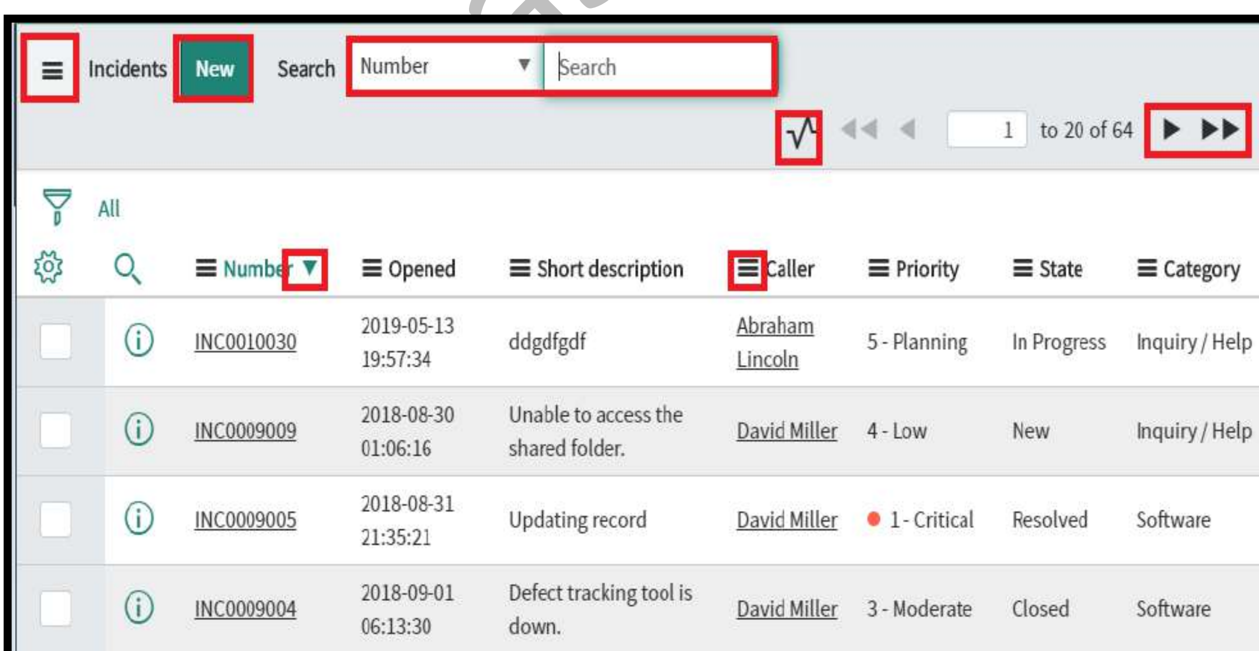
Users **can search, sort, filter, and edit data in lists**. Lists may be embedded in forms and may be hierarchical (have sub lists).

The list interface consists of a title bar, filters and breadcrumbs, columns of data, and a footer. Each column in a list corresponds to a field on the table.



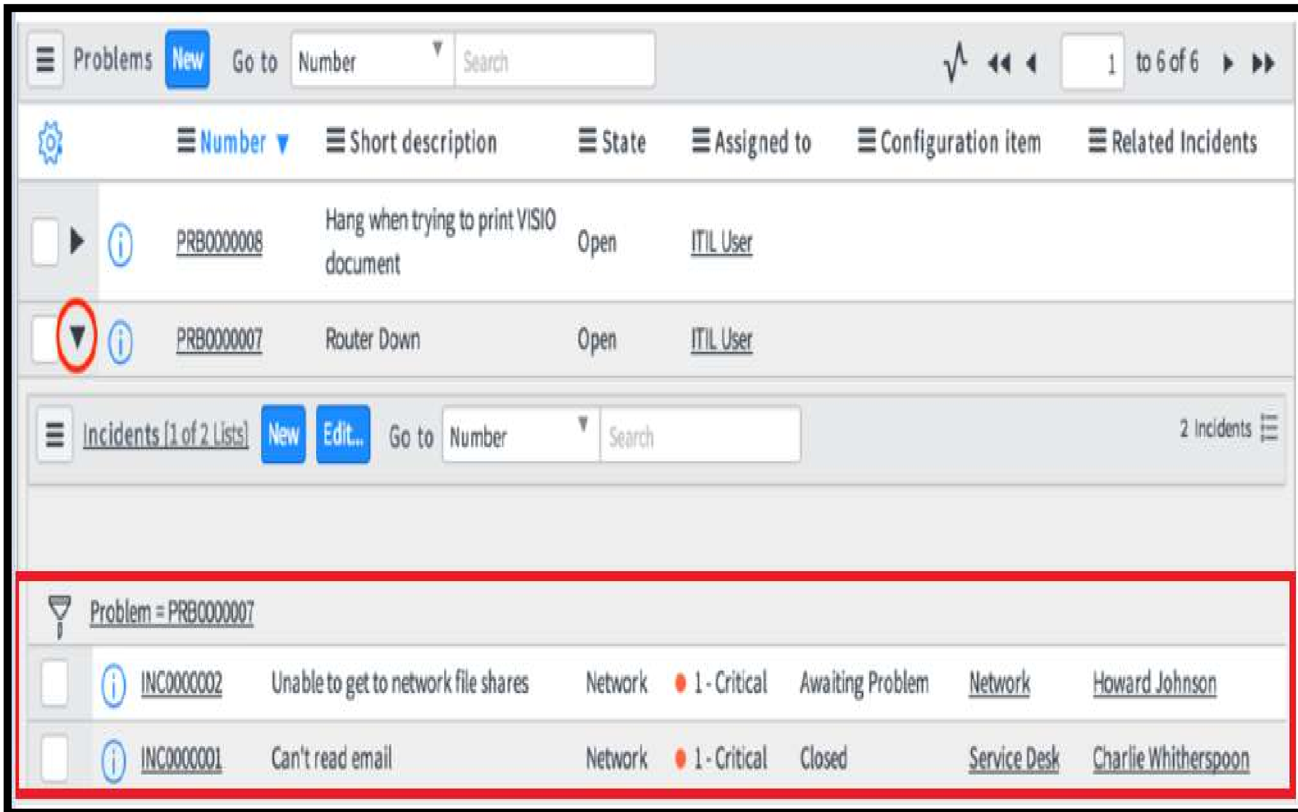
List properties and actions

The list interface consists of a title bar, filters and breadcrumbs, and columns of data. Each of these components provides features and lets you act on the list and the displayed records. There are two list



Hierarchical lists

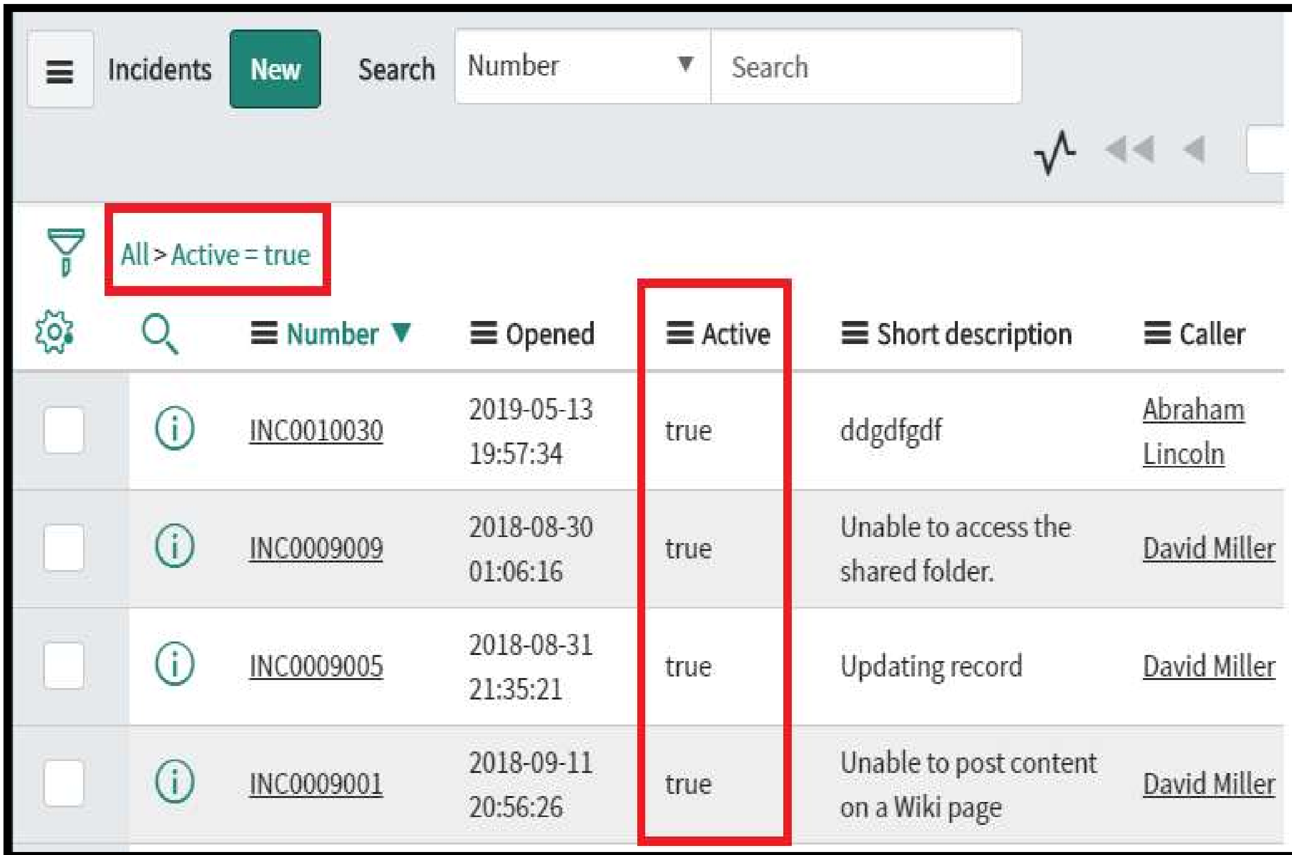
Hierarchical lists allow users to view records from related lists directly from a v2 list without navigating to a form.



Filters & Search

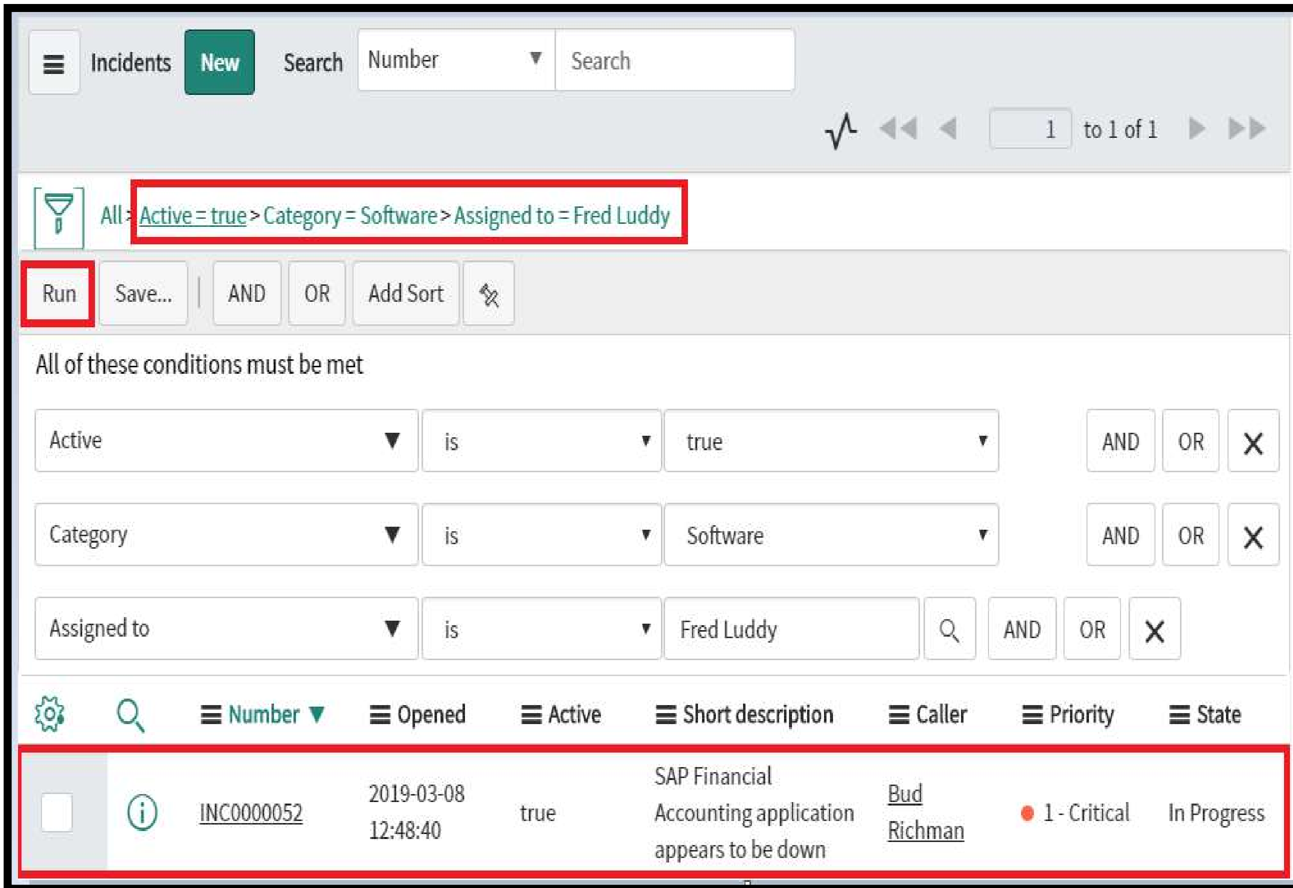
Filters can determine which table records are displayed in a list. When a developer creates a list module, the filter conditions are set. In the example, only records with the *Active* field value of *true* appear in the list. The syntax **All > Active = true** is known as a breadcrumb.

1. Many search bars
2. Similar search functions
3. Search wildcards



Note: Users can create or modify filters.

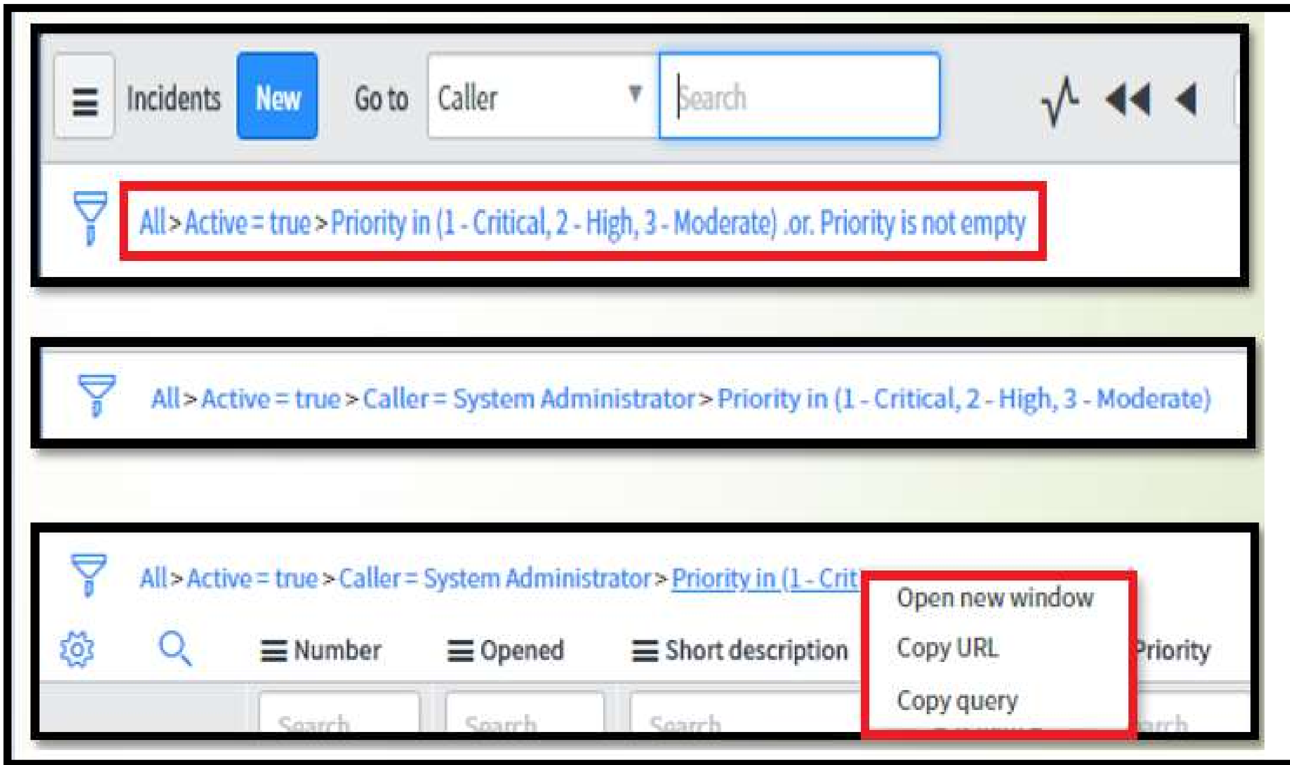
1. Click the Filter icon ( ) in a list. **(Condition Builder)**
2. Set the **[Field], [operator]**, and **[Value]**.
3. Use the **AND** button or the **OR** button to add additional conditions to the filter as required.
4. Click the Run button to apply the filter to the list. Only records meeting the filter condition appear in the list and the breadcrumbs are updated.



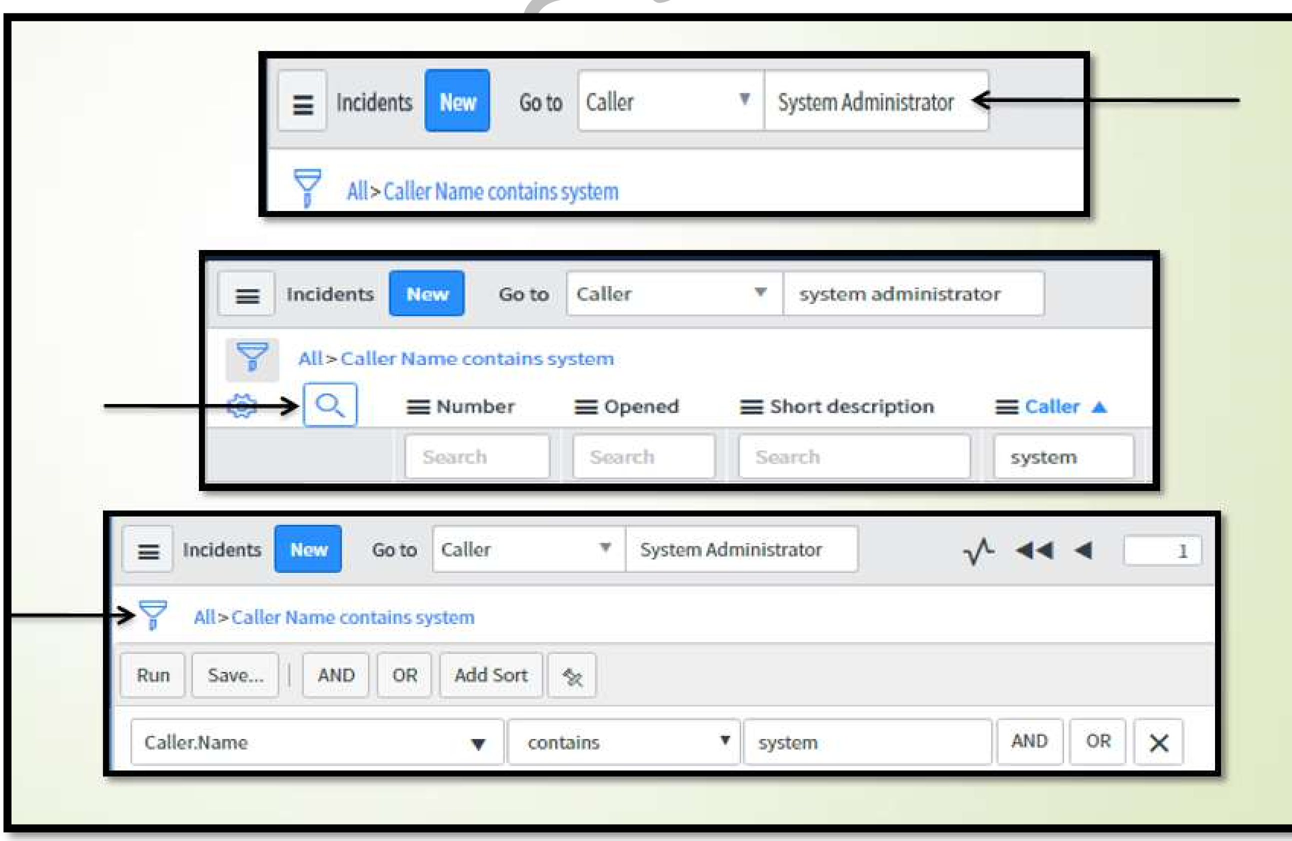
Breadcrumbs & Remove Condition

To remove a condition from the breadcrumbs, click **Remove next condition (>)** to the left of the condition to be removed. For example, to remove the *Active = true* condition, click the **>** between *All and Active = true.*

1. Dynamic
2. Remove a filter by clicking on **">"**
3. Copy query or URL

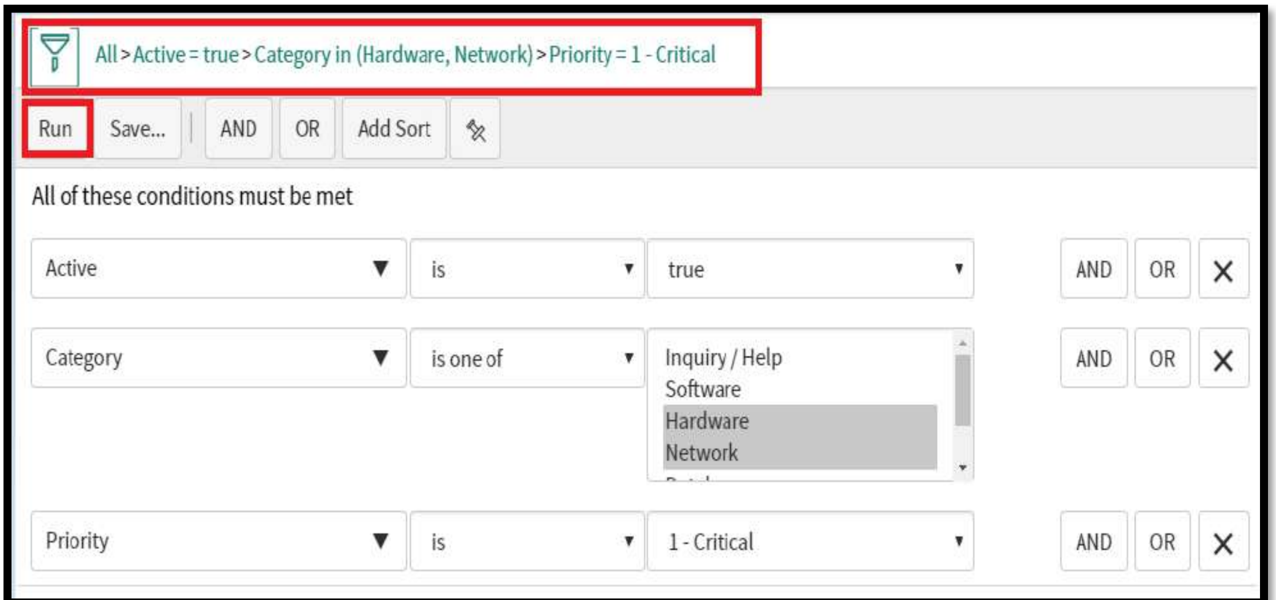


Apply Filter conditions in multiple ways

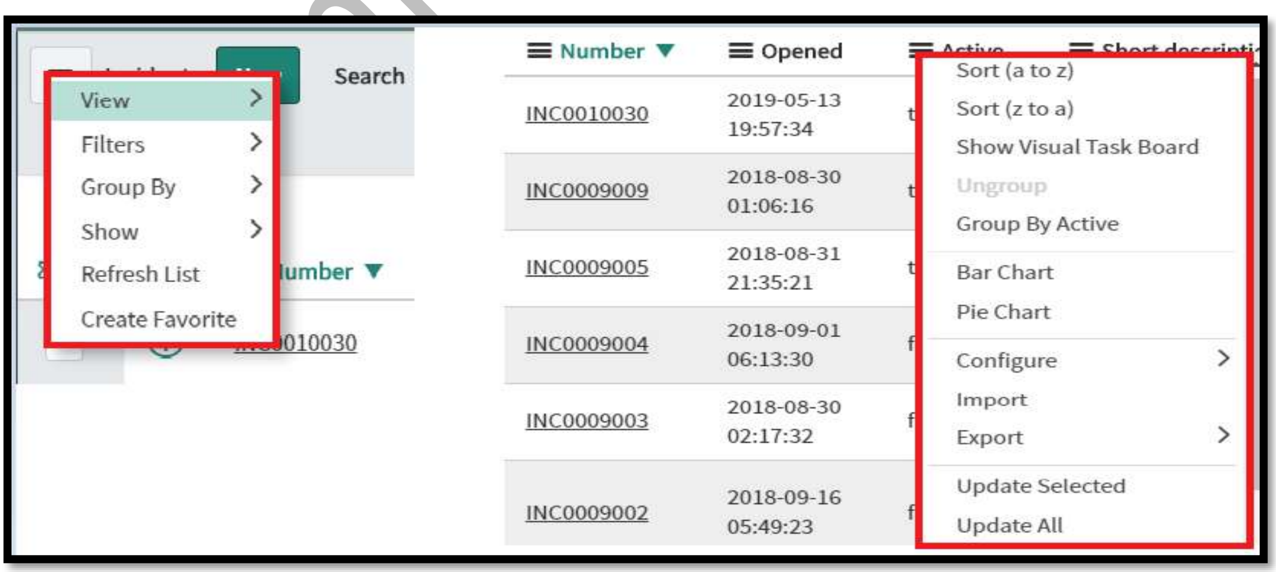


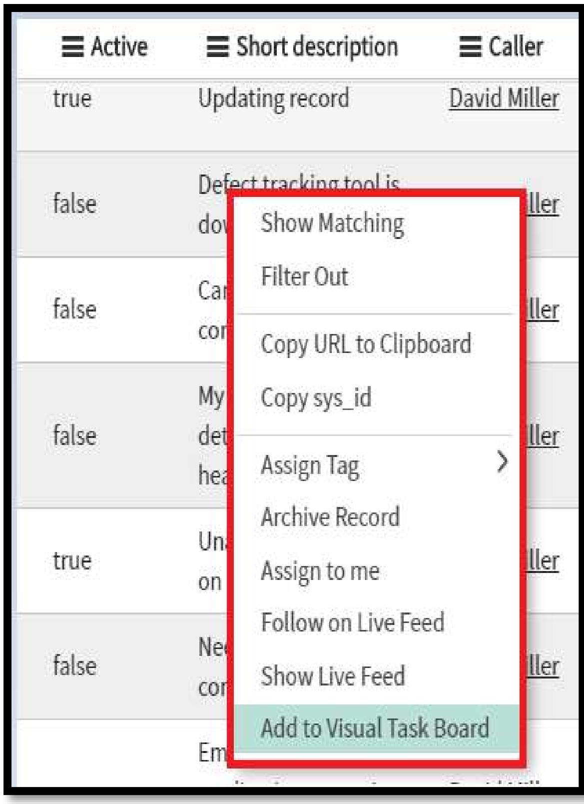
Condition Builder

1. Very powerful
2. Replaces SQL statements
3. Access related fields
4. Save condition
5. Copy condition
6. Field, Operator, Value



List Context Menus



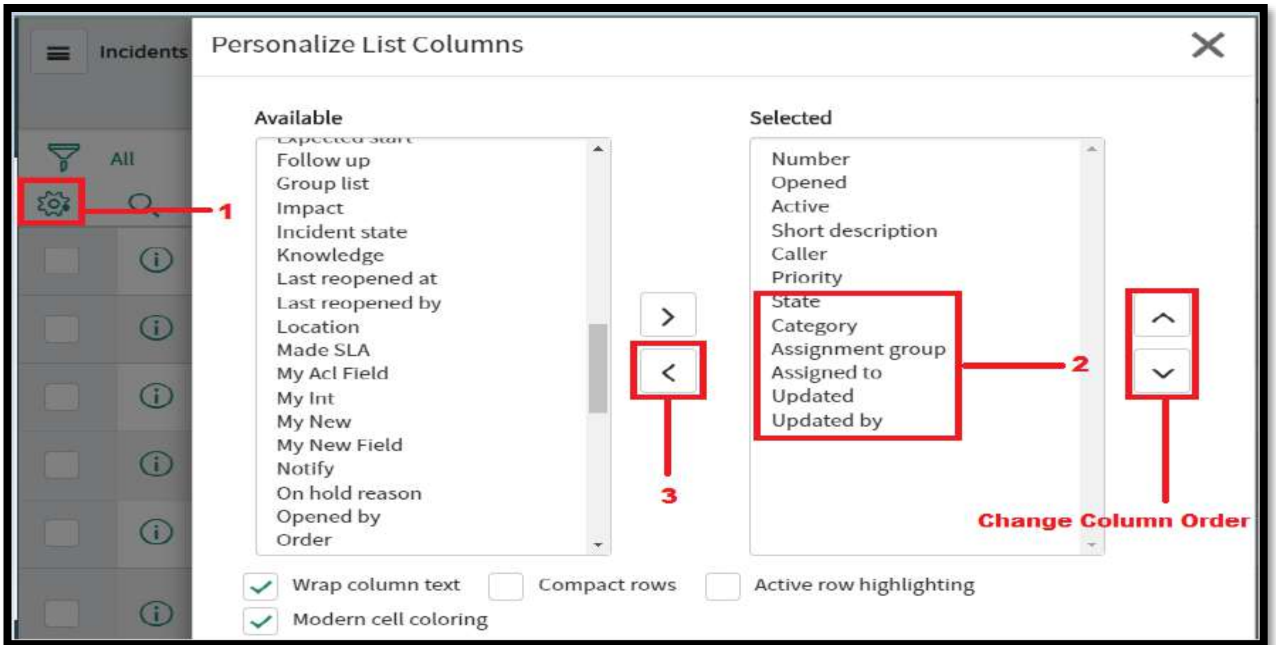


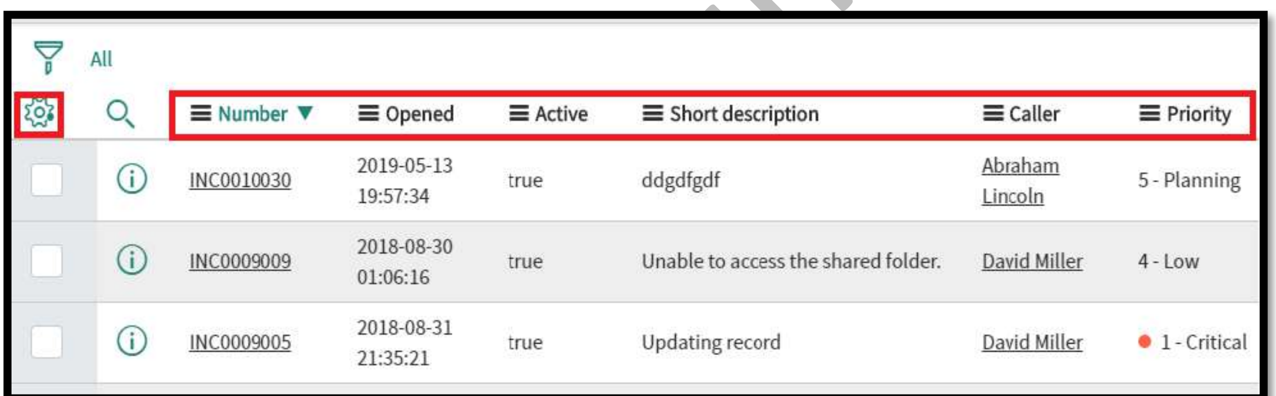
Personalize Versus Configure

Personalizing Stephen List View

1. **User specific**
2. **Add/remove any field on table**
3. **Reorder fields**

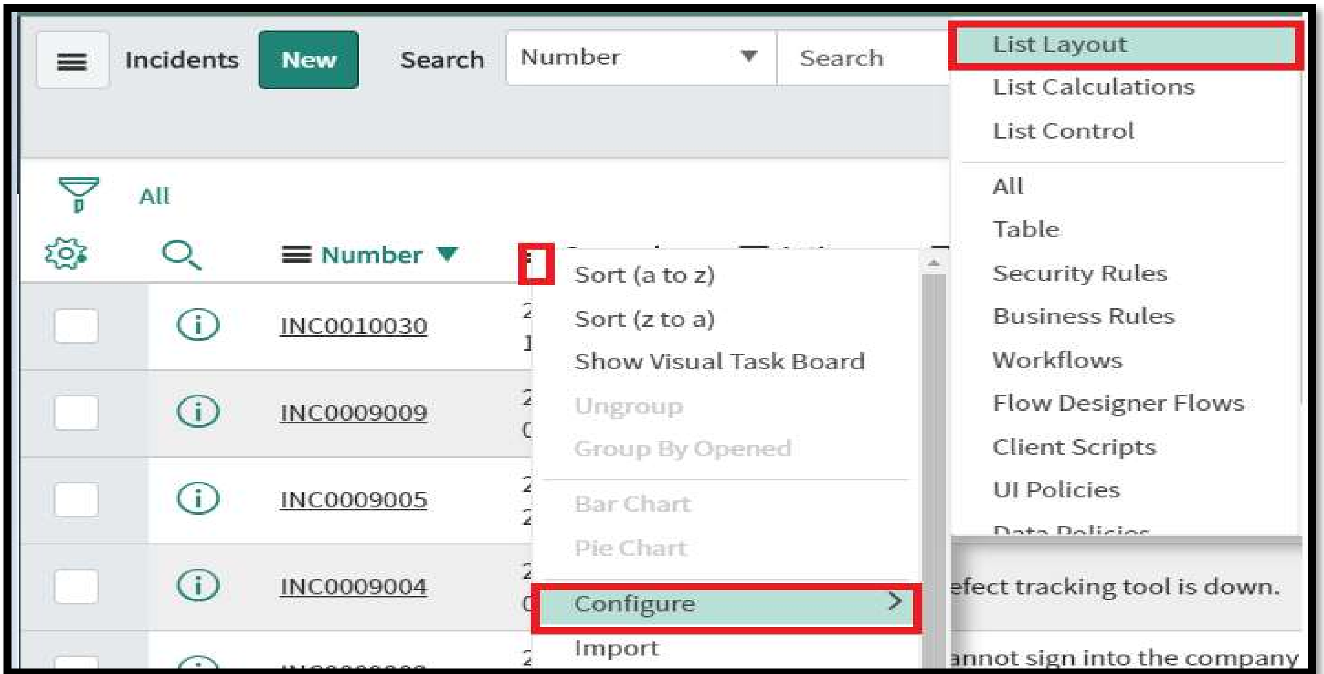
Customer Stefen

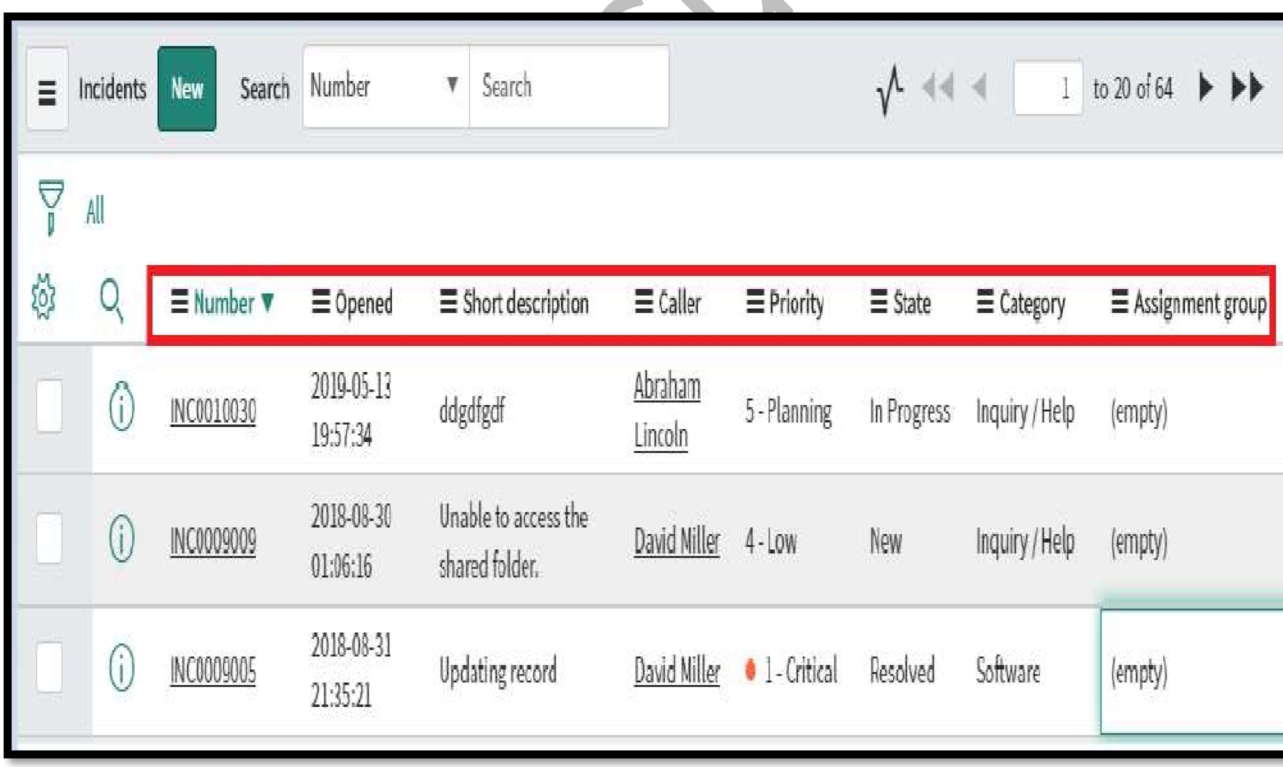




Configuring List Layout

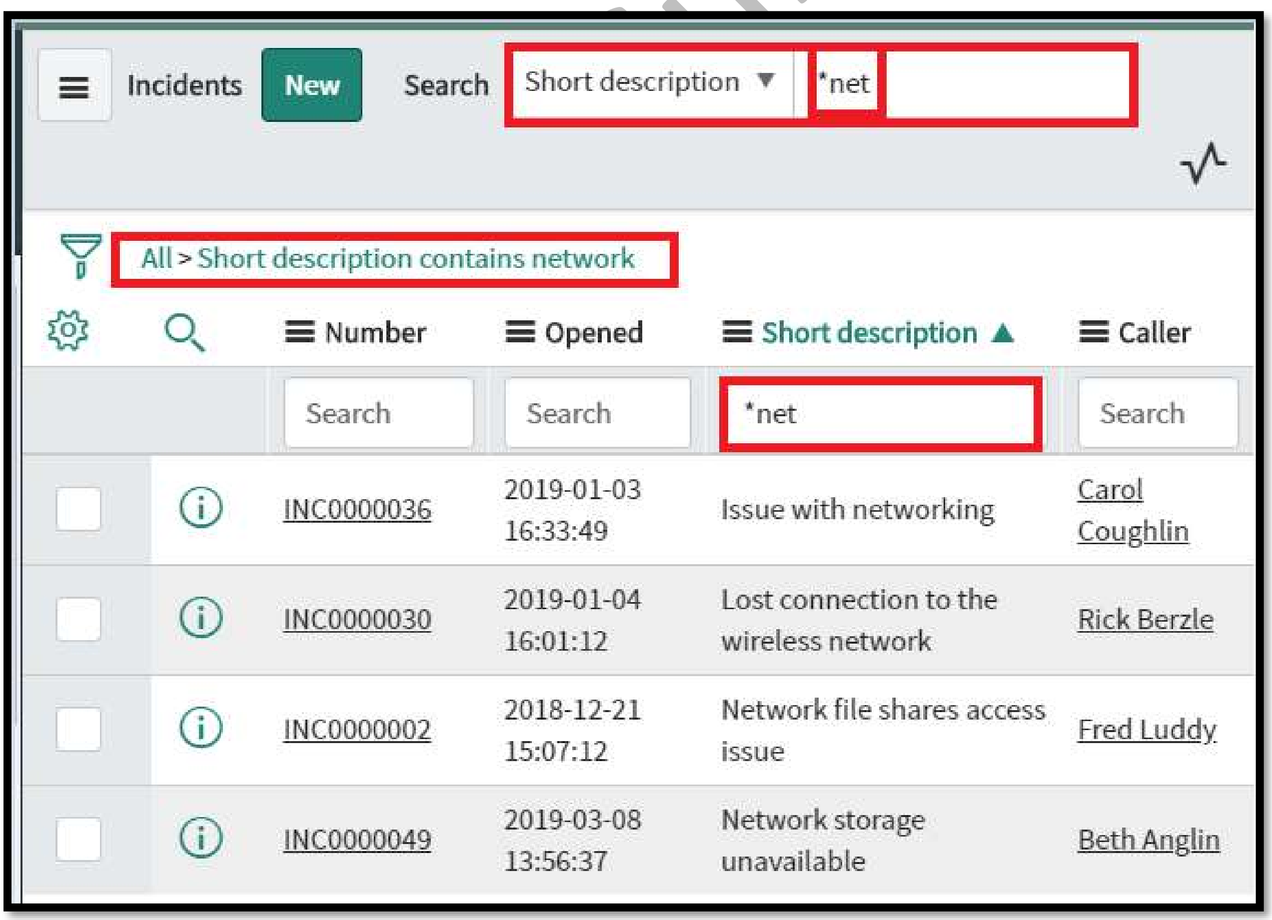
1. Different than gear icon
2. System wide
3. Permissions


Incidents
New
Search
Number
Search
List Layout
List Calculations
List Control
All
Table
Security Rules
Business Rules
Workflows
Flow Designer Flows
Client Scripts
UI Policies
All
Number
INC0010030
INC0009009
INC0009005
INC0009004
Sort (a to z)
Sort (z to a)
Show Visual Task Board
Ungroup
Group By Opened
Bar Chart
Pie Chart
Configure
Import

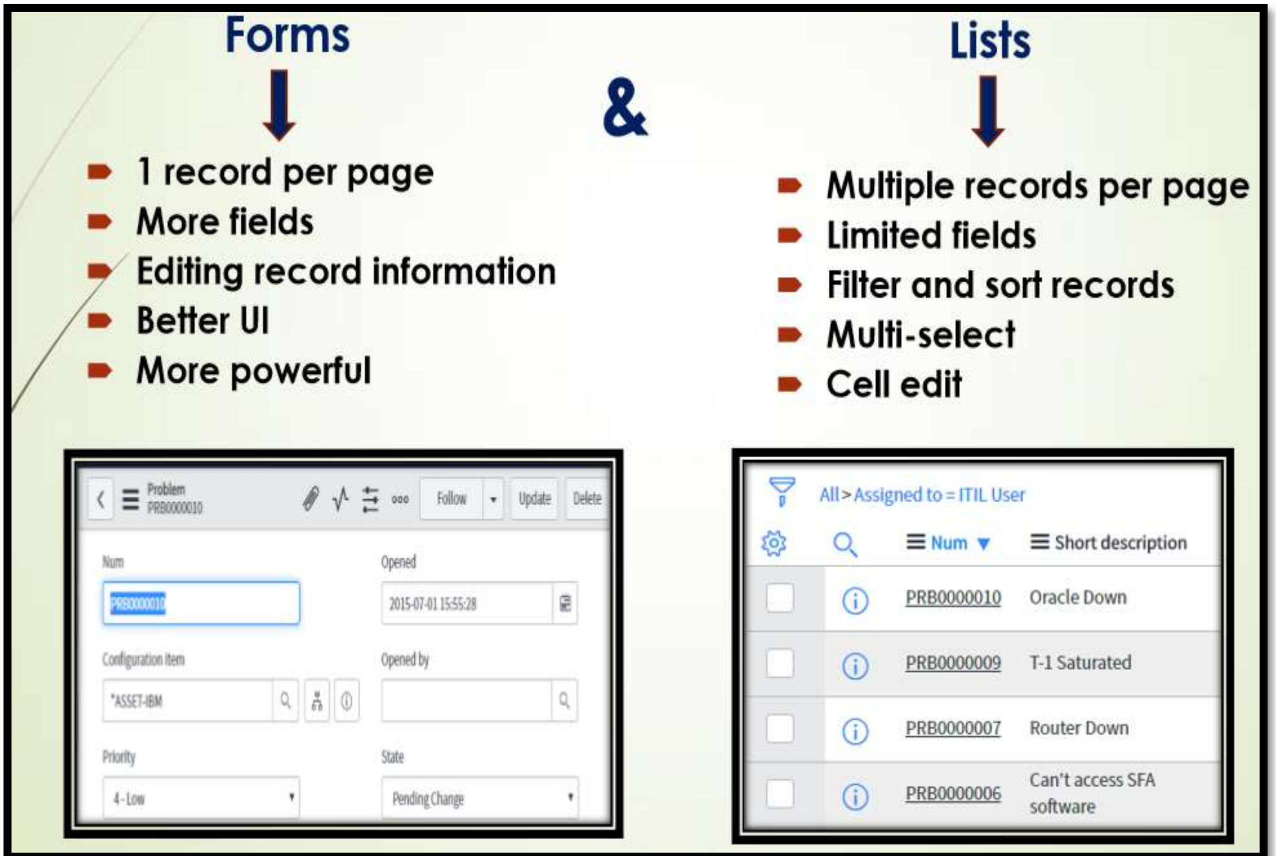

Incidents
New
Search
Number
Search
1
to 20 of 64
All
Number
Opened
Short description
Caller
Priority
State
Category
Assignment group
INC0010030
2019-05-13 19:57:34
ddgdfgdf
Abraham Lincoln
5 - Planning
In Progress
Inquiry / Help
(empty)
INC0009009
2018-08-30 01:06:16
Unable to access the shared folder.
David Miller
4 - Low
New
Inquiry / Help
(empty)
INC0009005
2018-08-31 21:35:21
Updating record
David Miller
1 - Critical
Resolved
Software
(empty)

Search Conditions with wild card Entries

Wildcard Syntax	Search Criteria	Example
*[term]	contains	*Mark
!*[term]	does not contain	!*Mark
=[term]	equals	=Miller
!=[term]	does not equal	!=Miller
[term]%	starts with	Hello%
%[term]	ends with	%goodbye



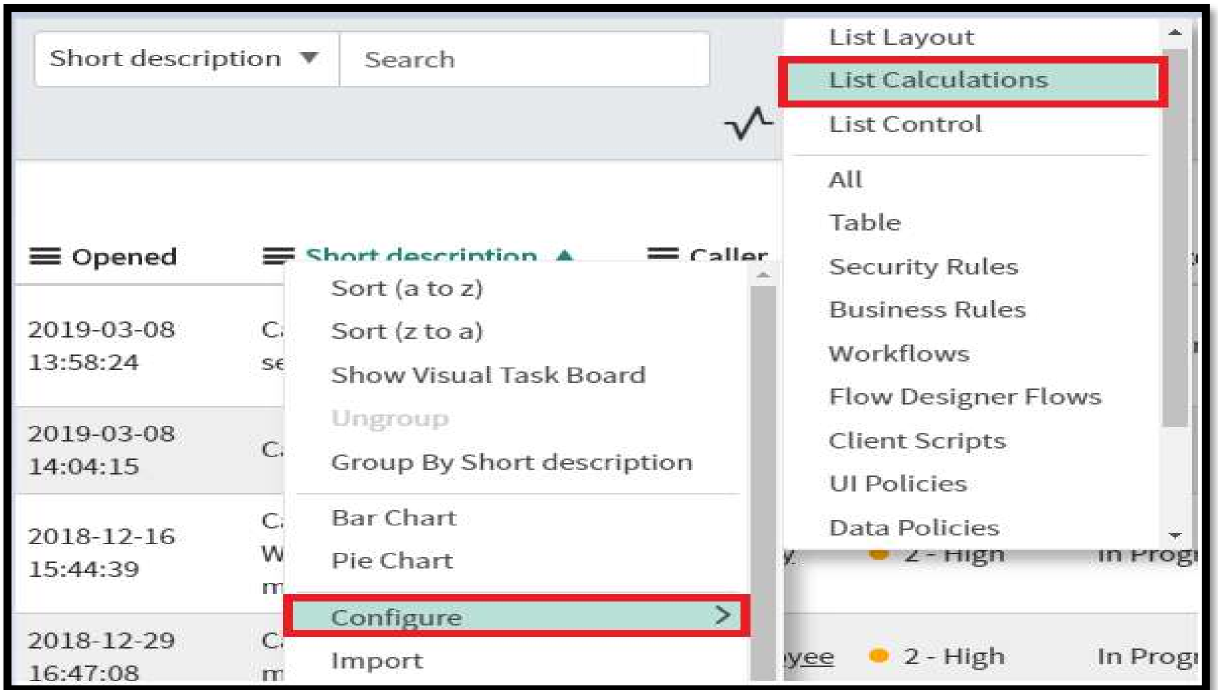
Difference between Lists & Forms

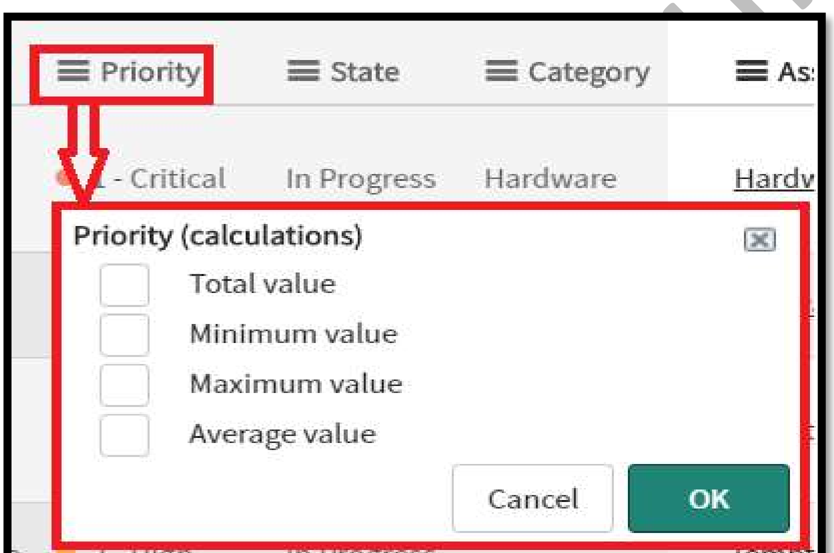


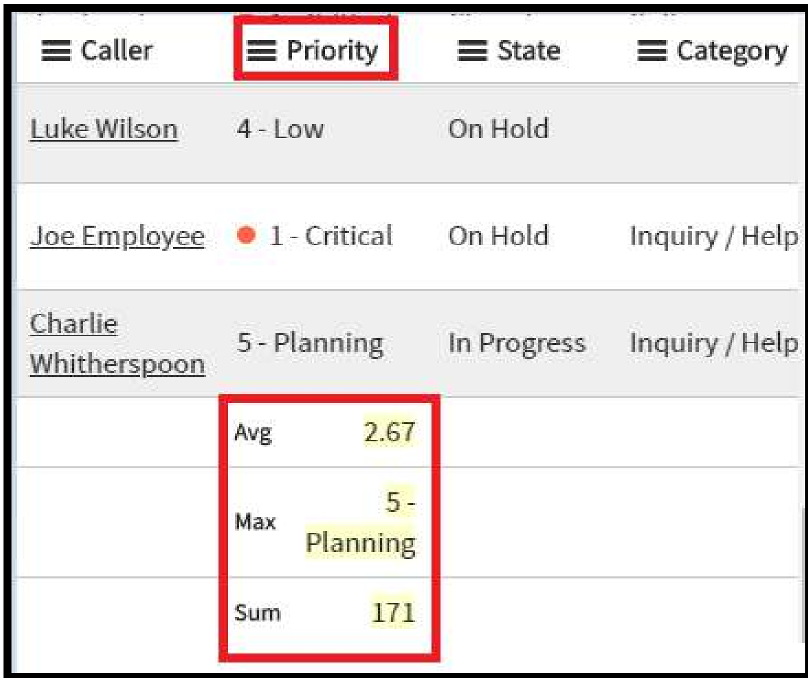
Configure List Calculation

You can configure list calculations for columns, which calculate column **totals, minimums, maximums,** and **averages.**

1. Select **Configure** > **List Calculations**.
2. The list calculation window appears.


Short description
Search
List Layout
List Calculations
List Control
All
Table
Security Rules
Business Rules
Workflows
Flow Designer Flows
Client Scripts
UI Policies
Data Policies
Opened
Short description
Caller
2019-03-08 13:58:24
2019-03-08 14:04:15
2018-12-16 15:44:39
2018-12-29 16:47:08
Sort (a to z)
Sort (z to a)
Show Visual Task Board
Ungroup
Group By Short description
Bar Chart
Pie Chart
Configure
Import
2 - High
In Progi


Priority
State
Category
- Critical
In Progress
Hardware
Priority (calculations)
Total value
Minimum value
Maximum value
Average value
Cancel
OK



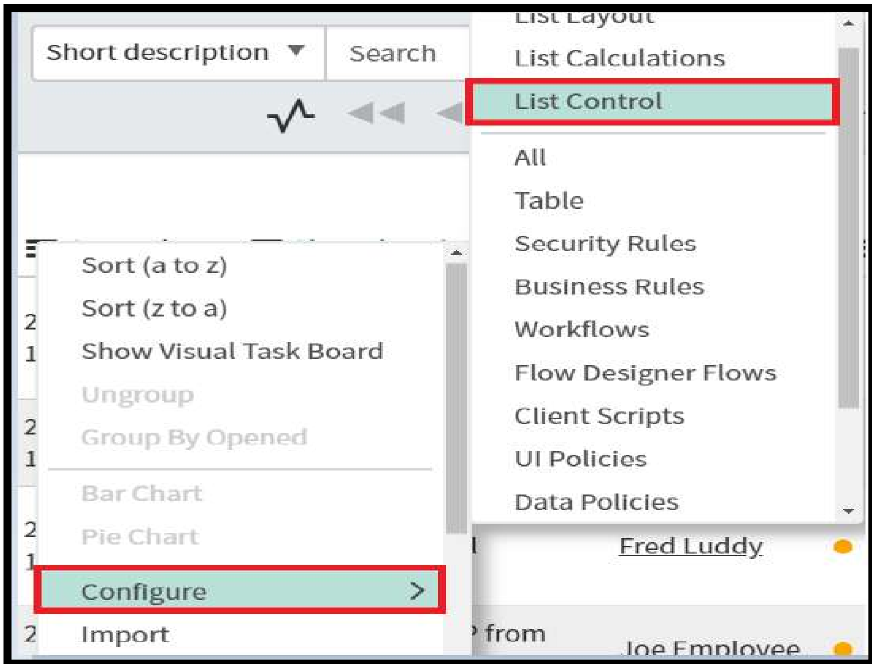
Note: List calculations are available in list reports.

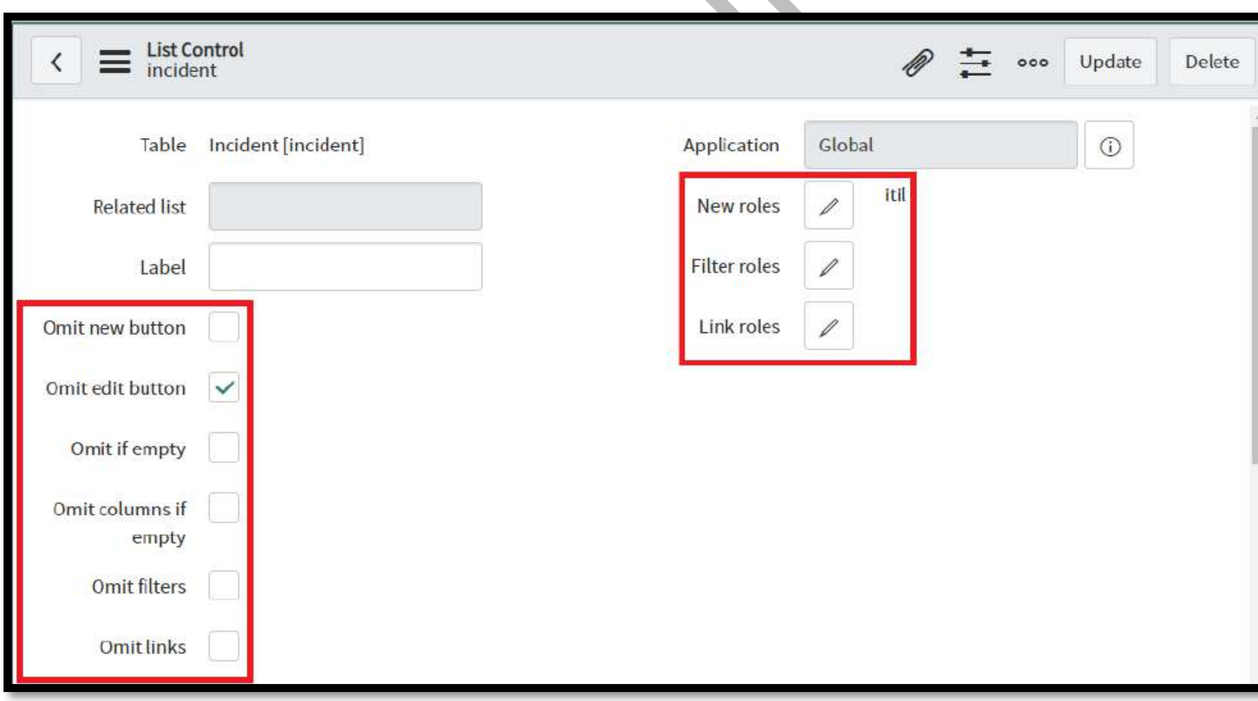
Configure list controls

We can configure list controls, such as buttons and filters.

Procedure

1. Open **Incident list.**
2. Open the list configuration page by performing the appropriate action for the list version


Short description
Search
List Layout
List Calculations
List Control
All
Table
Security Rules
Business Rules
Workflows
Flow Designer Flows
Client Scripts
UI Policies
Data Policies
Sort (a to z)
Sort (z to a)
Show Visual Task Board
Ungroup
Group By Opened
Bar Chart
Pie Chart
Configure
Import
Fred Luddy
from
Joe Employee


List Control
incident
Update
Delete
Table
Incident [incident]
Application
Global
Related list
Label
New roles
itil
Filter roles
Link roles
Omit new button
Omit edit button
Omit if empty
Omit columns if empty
Omit filters
Omit links

Chapter - 6

Plugins

What is Plug-ins

1. Service-Now provides functionalities or features in the form of system modules. Usually,
2. Modules are activated by default, but if you want to add additional functionality,
3. Then you can consider system plugins. Plugins are like small software components that Provide features and functionality as an option. Let's better understand this.

Example of Plugins

1. **Example:** Data Archive is a plugin that is not activated by default.
2. The size of the data Increases in Service-Now according to the time that has passed
3. For example, an incident **More than a year old** is less significant as compared to a current incident, and old data may
4. Affect the performance of **queries and reports.**

How plugins are activated

Plugins may be added to the base platform in one of the following ways.

1. Activated by default in the base Service Now platform
2. Plugins are activated by an administrator (not by default)
3. Requested from Service Now before activating. This may be the case when a plugin is new, is appropriate for only certain deployments, and/or can only be activated by Service Now personnel.
4. Plugins that are associated with a for-fee subscription to an application or suite can be activated only after your organization has purchased the subscription.
5. Plugins may contain demo data for better understanding about specific application
6. **1000** of plugins are available in service now

After a plugin is active, you cannot disable or deactivate it. You can hide the functionality, if needed

Importance in plugins relationship

Some plugins depend on other plugins being activated.

In some cases, a plugin cannot work at all unless one or more other plugins are installed. When you ask to activate a plugin in this category, the system notifies you of the dependencies before it completes the activation. If you ask to continue, the system automatically activates all the plugins required.

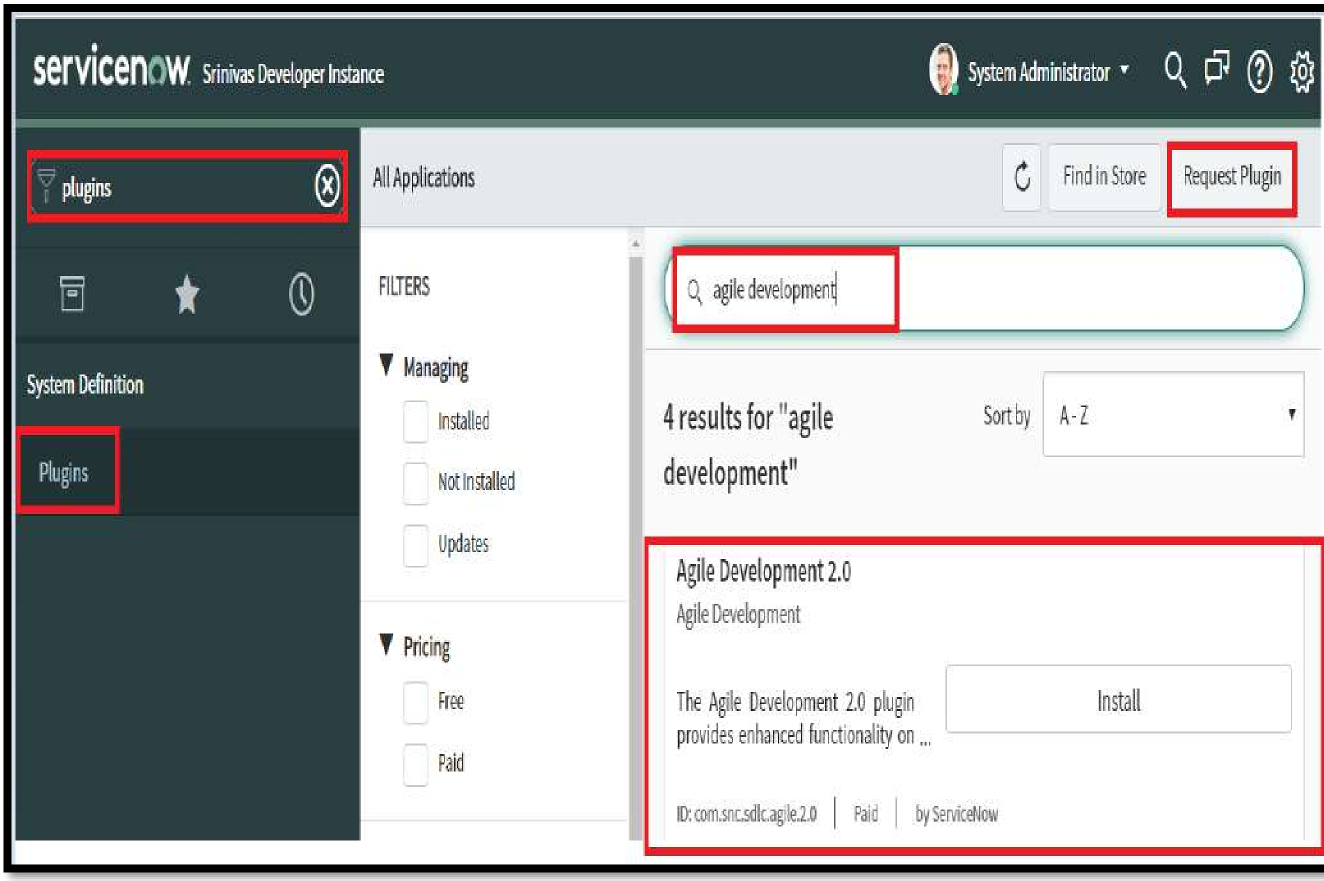
In some cases, the plugin works, but certain features that do not apply to your system are not installed. When you ask to activate a plugin in this category, the system notifies you of the dependencies before it completes the captivation. If you ask to continue, the system activates the requested plugin but does not include the features that depend on other plugins.

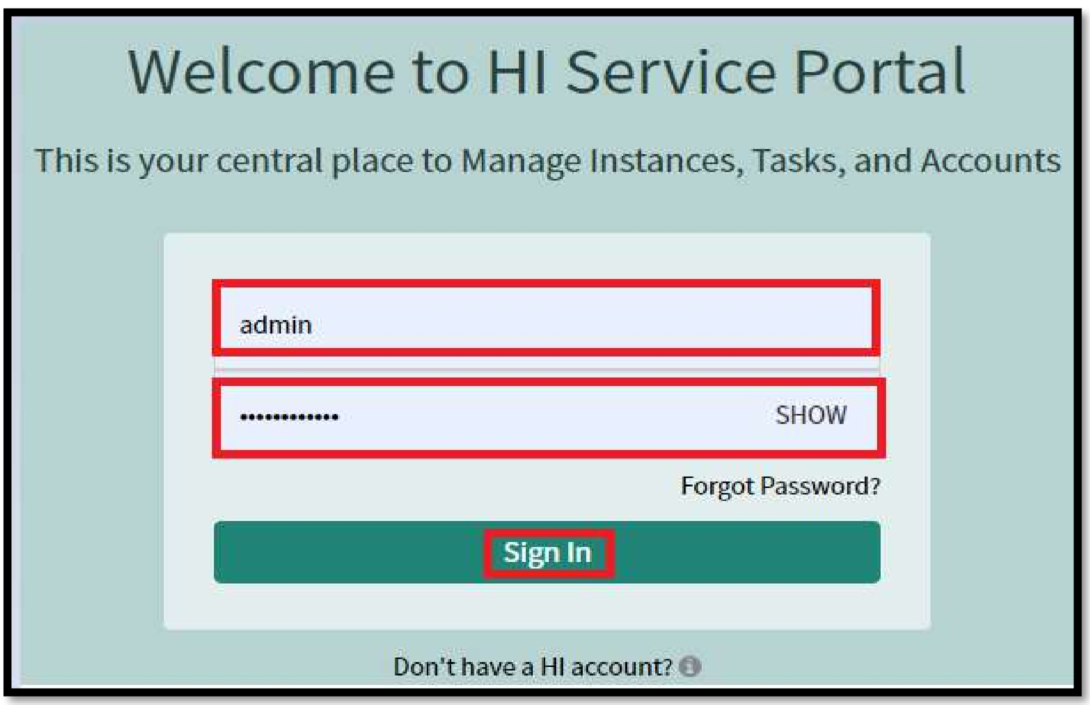
How to request a plugin?

Some plugins should be activated by default from Service Now personnel. These plugins do not appear in the **System Definition** > **Plugins** list.

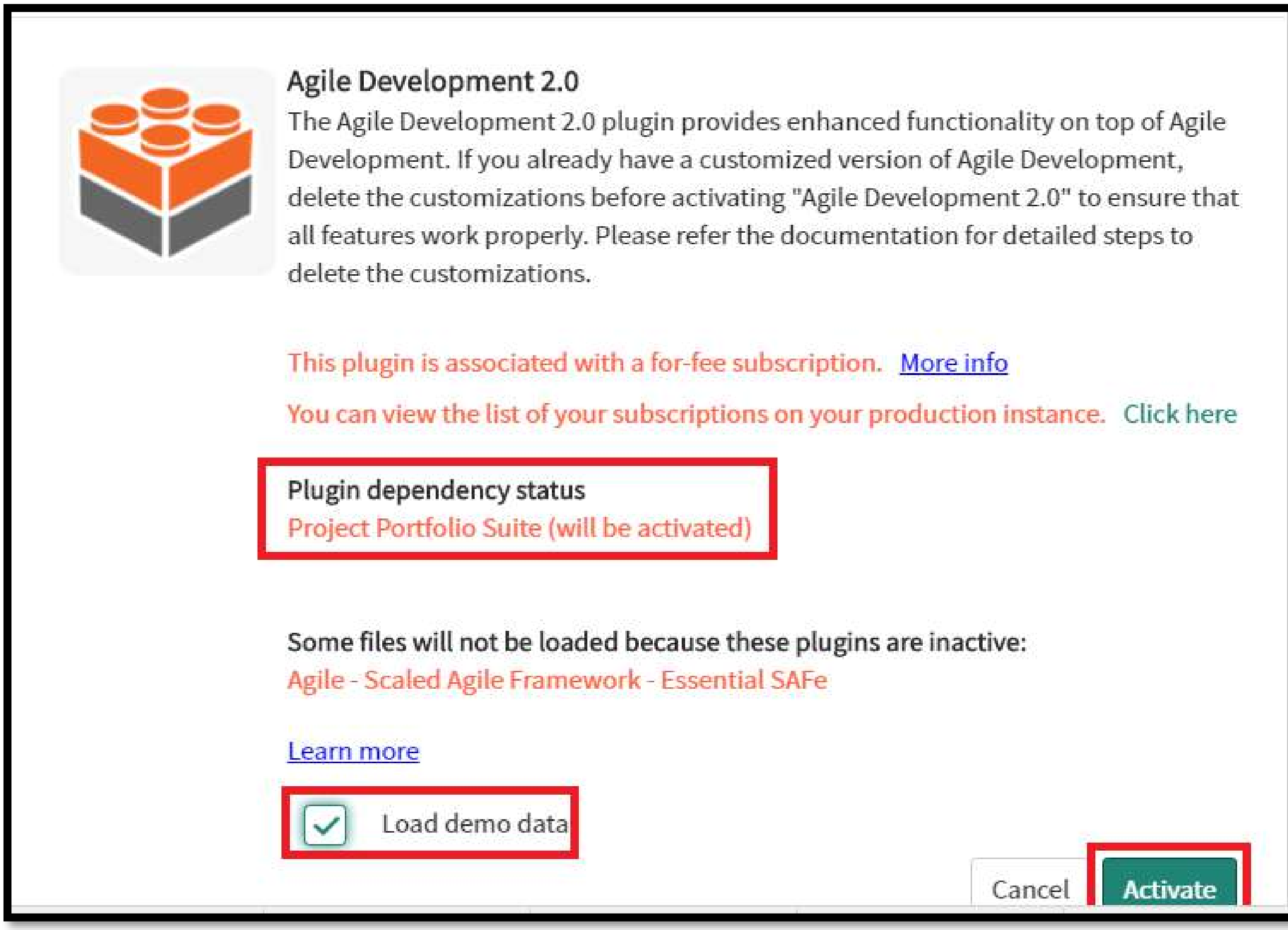
Procedure

1. In the **HI Service Portal**, click **Service Requests** > **Activate Plugin**.
2. Fill out the form.

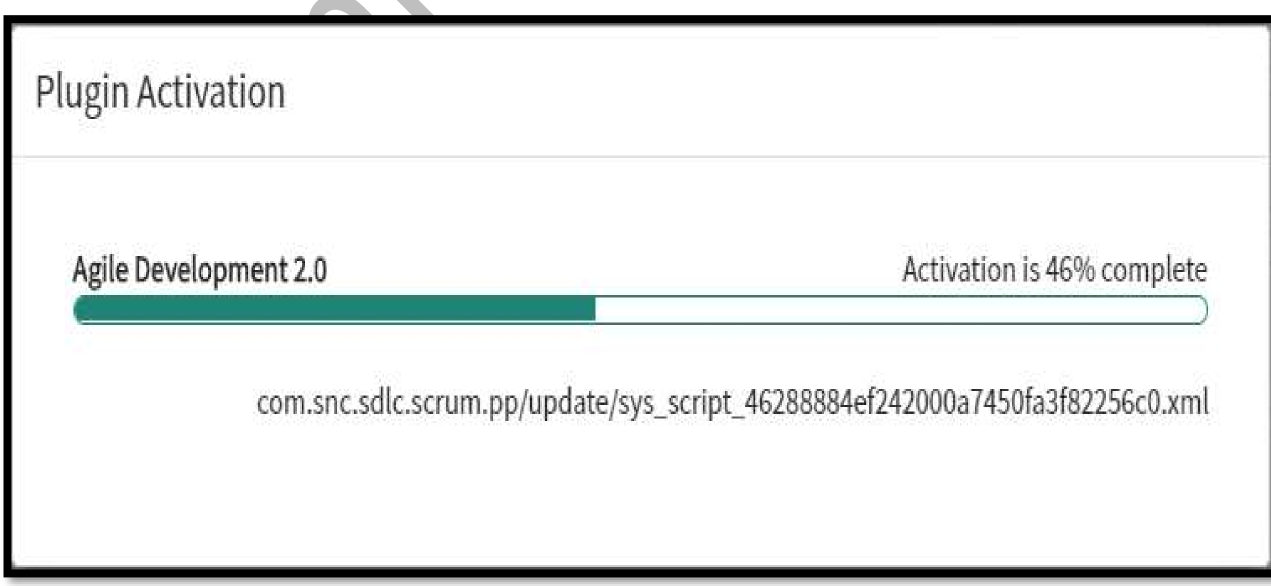


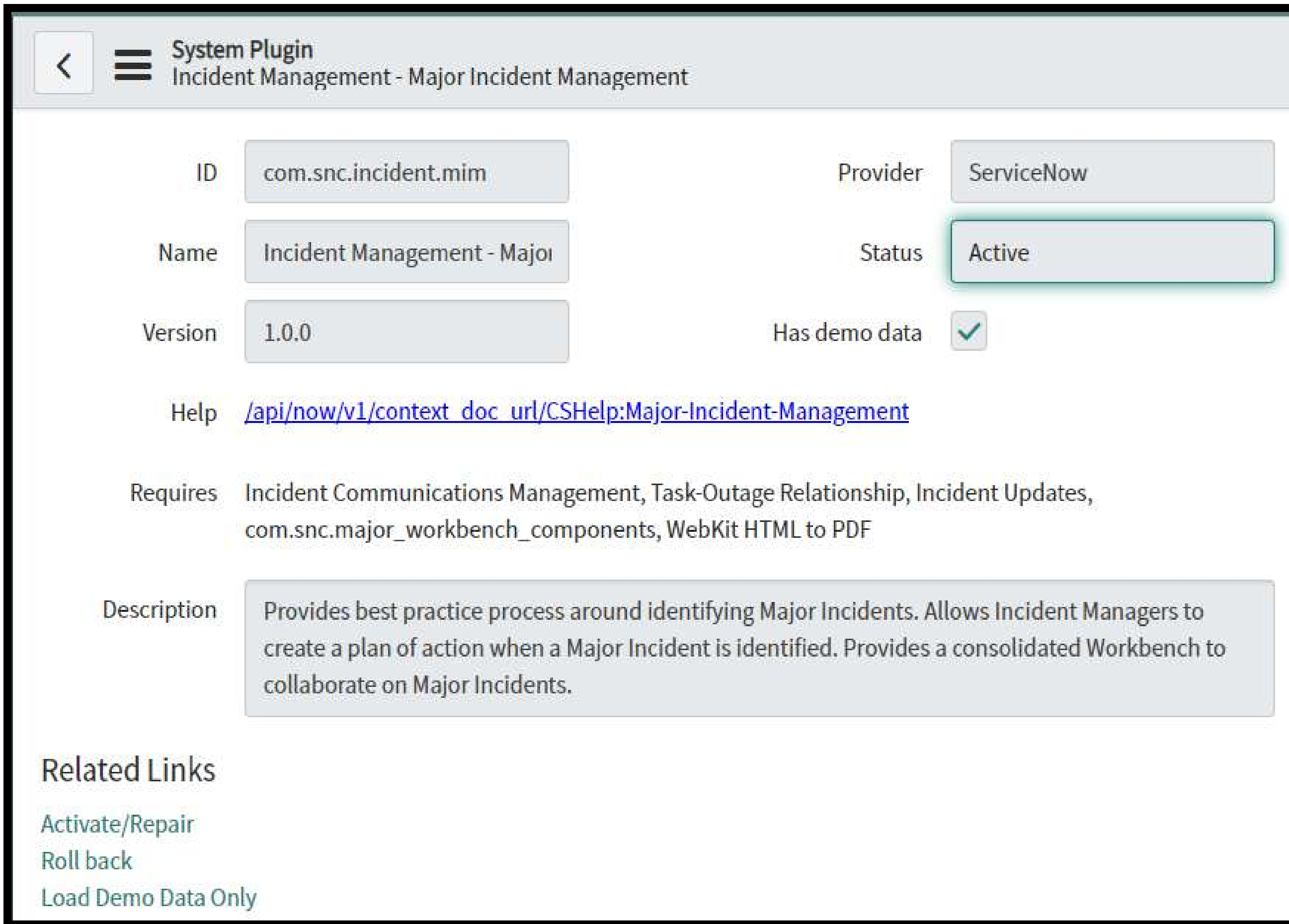


3. After subscribe the plugin, we can activate in our instance

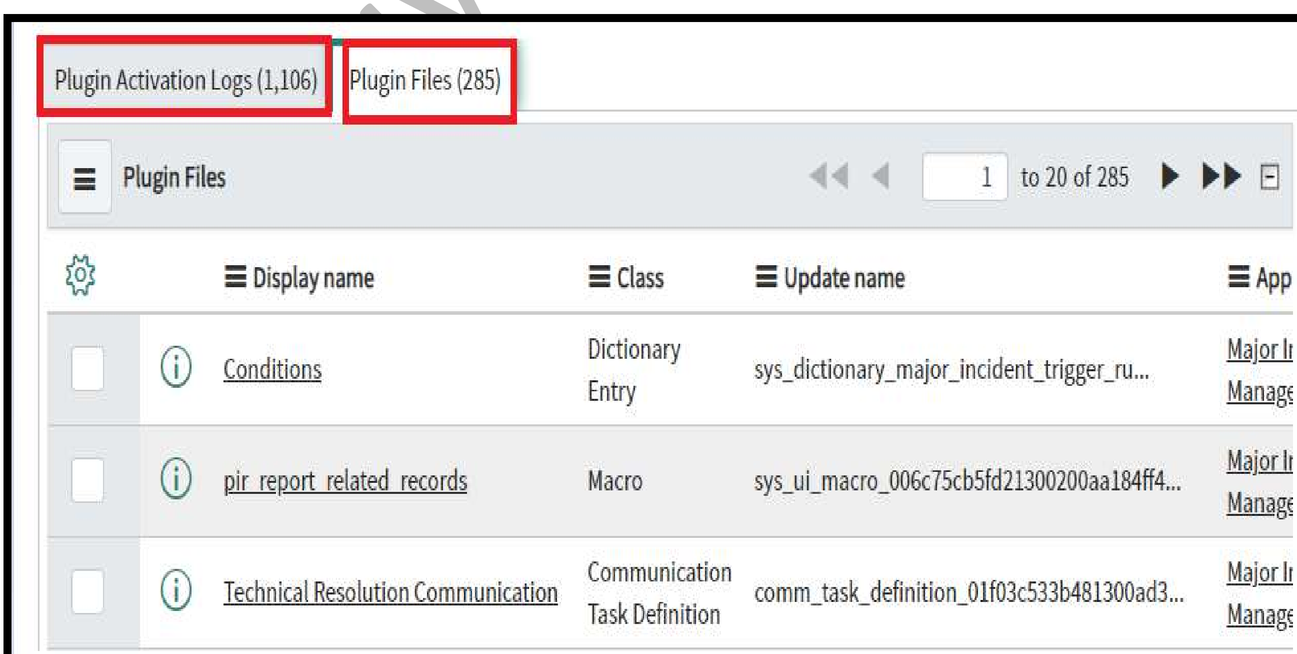


4. Before activating the plugin enable check box of **Load demo data**
5. Click on **Activate** button
6. Take few minutes of time to activating plugin (Track the status like below)





7. Here we can able see plugin details



Purchase a plugin

Some features require a separate subscription from the rest of the Service Now platform.

To purchase a subscription, we need to contact your Service Now account manager. In most cases, the account manager will arrange to have the plugin activated on your organization's production and non-production instances, generally within a few days. In some cases, you can activate the plugin within the instance.

If you do not have an account manager, decide to delay activation after purchase, or want to evaluate the feature on a non-production instance before purchase, follow the steps to **activate a plugin.** If the plugin is not listed in the **System Definition** > **Plugins** module, make a **request through HI Service Portal.**

Chapter – 7

Tables

What is Table

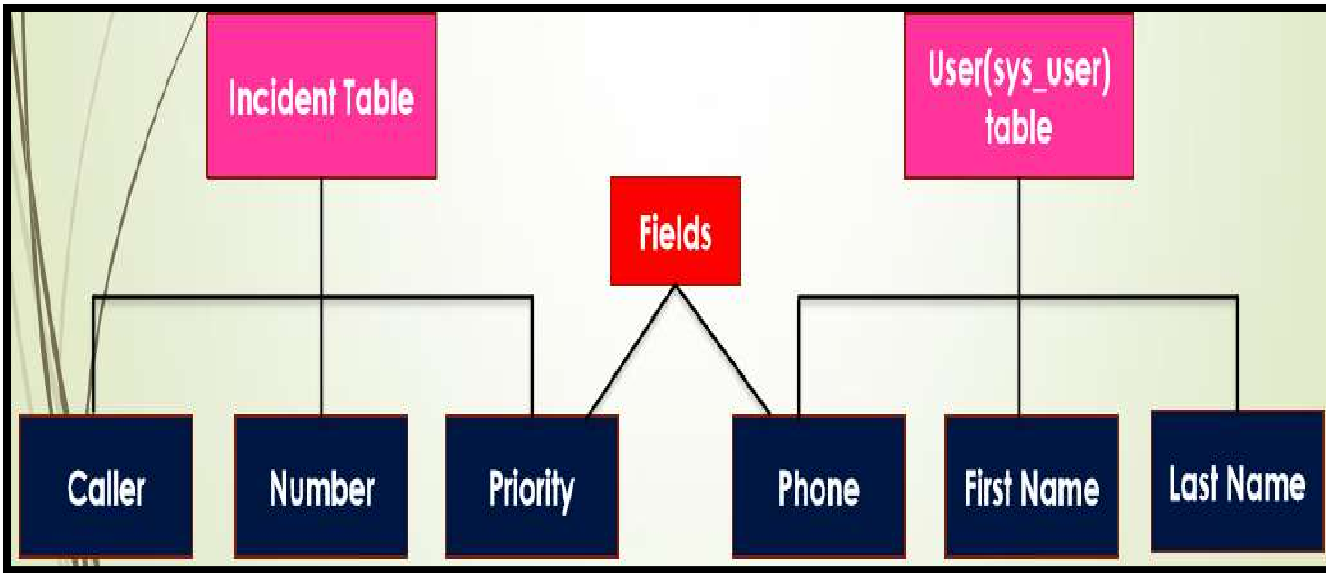
A table is a collection of records in the database. Each record corresponds to a **row in a table**, and each field on a record corresponds to a **column on that table**.

Applications use tables and records to manage data and processes, such as **Incident, Problem, Change_request** and **CMDB.** Tables can extend other tables, creating parent tables and child tables.

1. **Tables module:** Provides a list of all tables in the database.
2. **Tables & Columns module:** Provides a list of all existing tables, with columns, column attributes, and indexes.
3. **Schema map:** Provides a graphical representation of the relationships between multiple tables.
4. **Data dictionary tables:** Contains additional information that defines database elements.

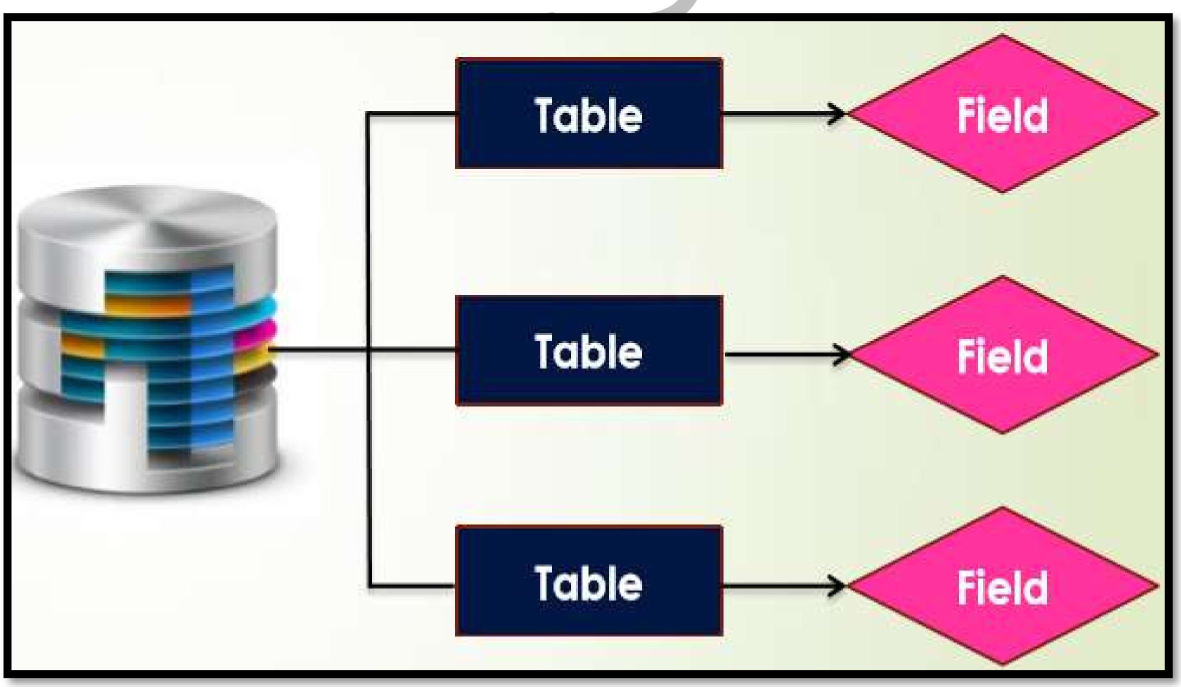
Database Table Overview

1. Over **2,000** tables in base instance
2. Each table has many **Fields or Attributes**
3. One Table can **extend** to other tables
4. Naming convention
5. Admin can **Create/Modify** tables
6. Each application has **1 or many** tables

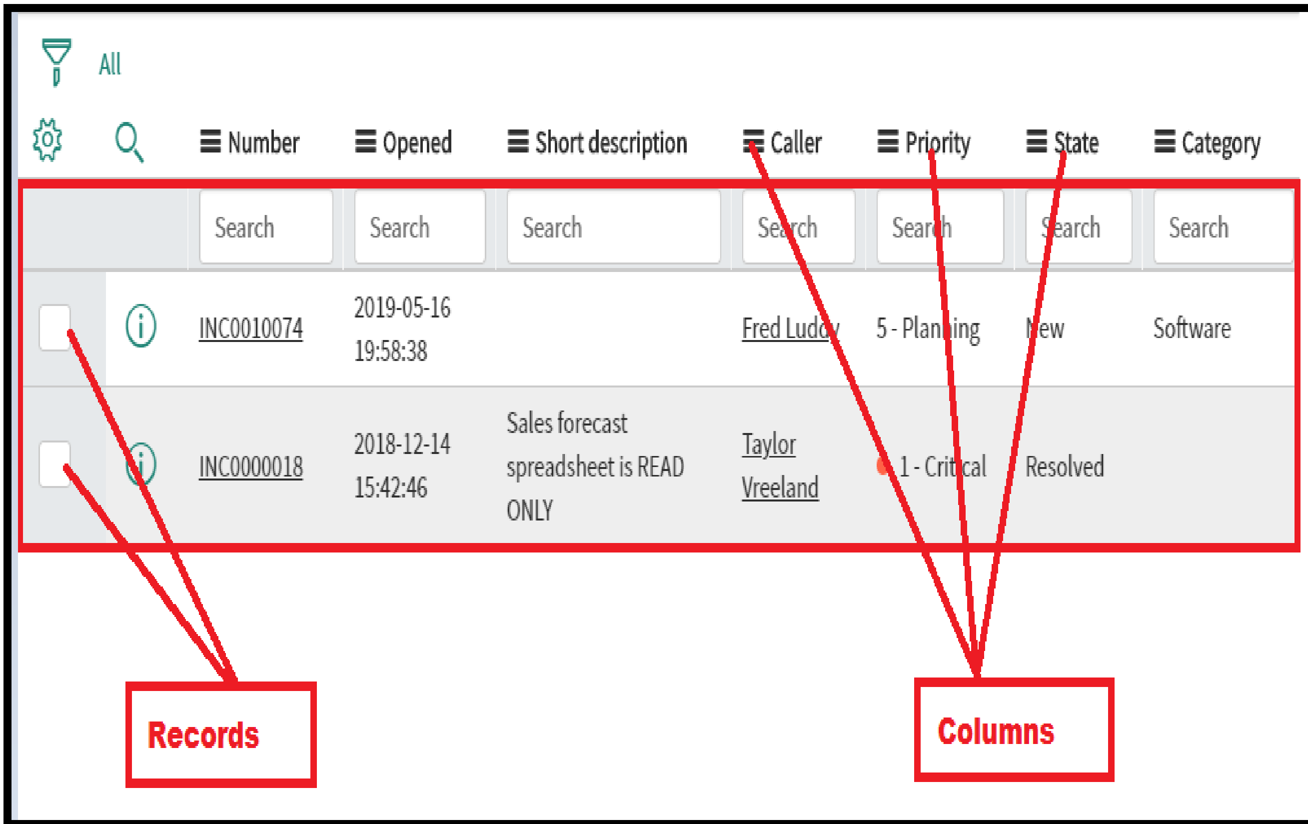


Database, Tables, & Fields

1. A database contains many **tables**
2. A table contains many **fields**
3. Records are stored in **tables**



4. Each **Row** can be representing a one **Record** in the table
5. Each **Field** can be representing a one **Column** in the table



Service Now Table Relations and Classes

We can enable one or more child tables to share fields and records with a parent table. Administrators and Service now developers can **only extend tables during the table creation.**

Parent Table	Child Tables
Task	**Incident**
	Problem
	Change_Request
[cmdb_ci]	**cmdb_ci_computer**
	cmdb_ci_database
	cmdb_ci_appl

Types of tables

Base table:

Base tables are the tables already available in Service Now and which do not extend any table

We are calling with following names

1. Extended Table
2. Base Table
3. Parent Table
4. Out of the box Table

Custom table:

A table is custom if it was created by an administrator and is not part of a **system upgrade or plugin activation.**

Major Tables in Service Now

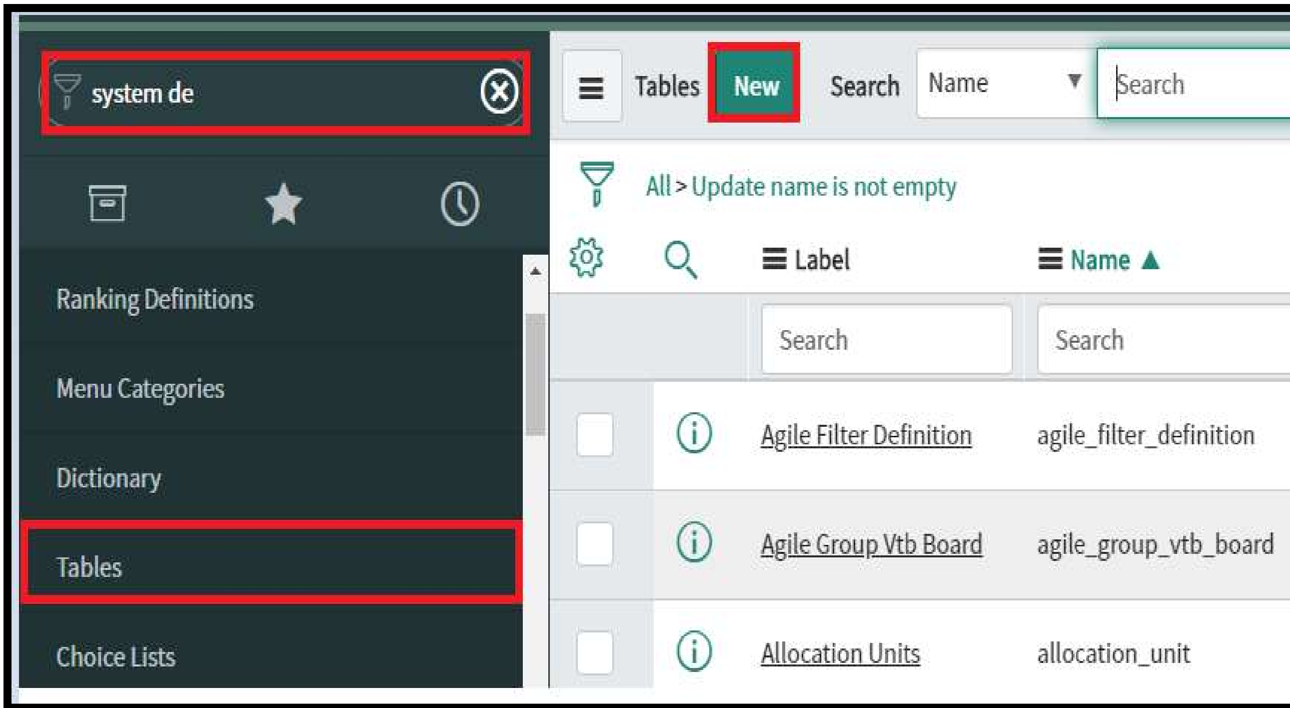
Major Tables	
✓ Task	✓ kb_knowlege
✓ incident	✓ kb_category
✓ problem	✓ kb_knowledgebase
✓ change_request	✓ sc_catalog
✓ sys_user	✓ sc_cat_item
✓ sys_user_group	✓ sc_req_item
✓ sys_user_role	✓ cmdb_ci
✓ cmn_location	✓ cmdb_computer

Exercise: 1 How to create a custom table

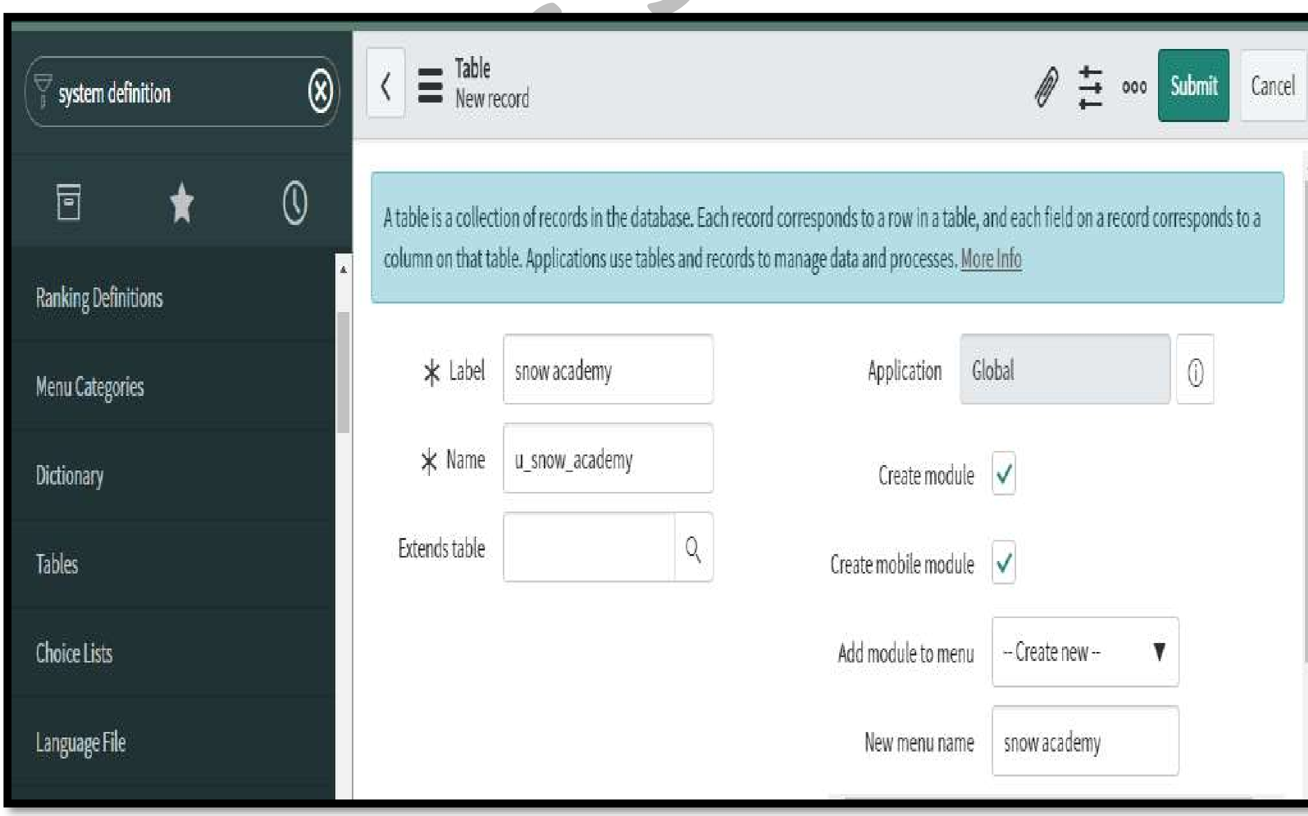
Service Now Administrators and service Now application developers can create custom tables to store application data.

Procedure

1. Navigate to **System Definition > Tables.**
2. Click on **New**



3. Fill the required details in form
4. Click on **Submit**



Global default fields

When we create a new custom table in service now platform, several system fields are appearing in the Table Columns embedded list. For all tables, required system fields are added automatically. **We cannot delete or modify these fields.**

For tables that extend another table, fields on the parent table also appear on the Table Columns embedded list for the current table. If you modify these fields, remember that all changes to fields on the parent table also affect all child tables, not just the current table.

These required system fields are added to all tables:

Default system fields

1. Class **[sys_class_name]**
2. Created **[sys_created_on]**
3. Created by **[sys_created_by]**
4. Sys_id **[sys_id]**
5. **[sys_mod_count]**
6. Updated by **[sys_updated_by]**
7. Updated **[sys_updated]**

Exercise: 2 Delete a table

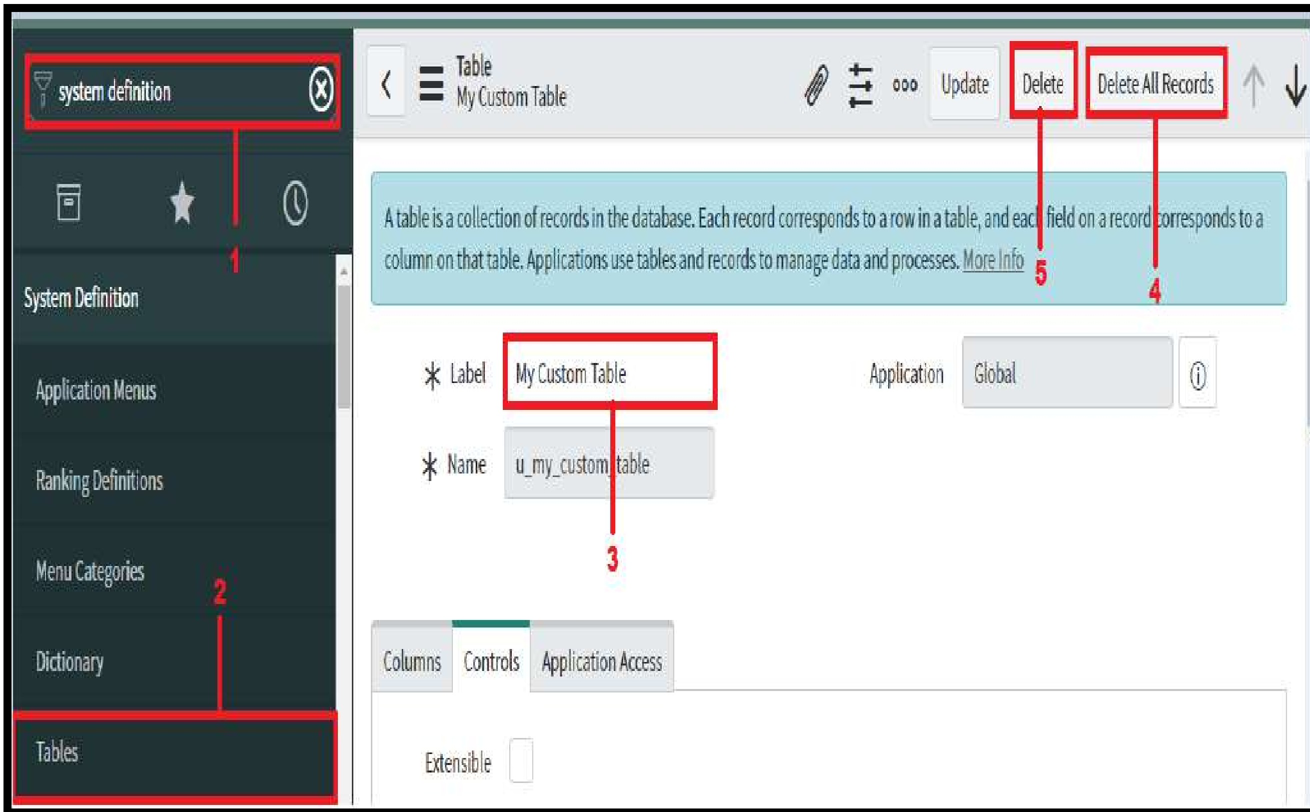
Only administrators can be able to delete a custom tables that are no longer required. For example, delete a table from an application that is under development because the business requirements change.

A table is custom if an administrator created it and it is not part of a system upgrade or plugin activation. Custom table names always begins with **u_, or x_ for scoped tables**.

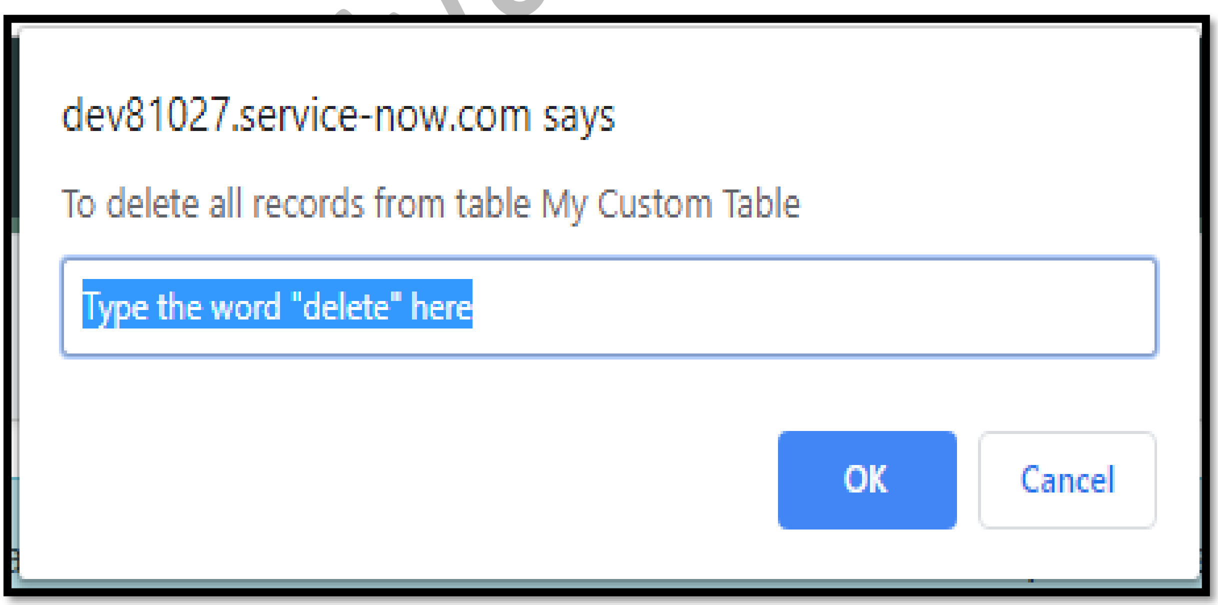
Warning: We can delete the records in the table before we delete the table. If you do not delete the records first, errors can result on the parent table if it references the records. Deleting the records removes any references from the parent table.

Procedure

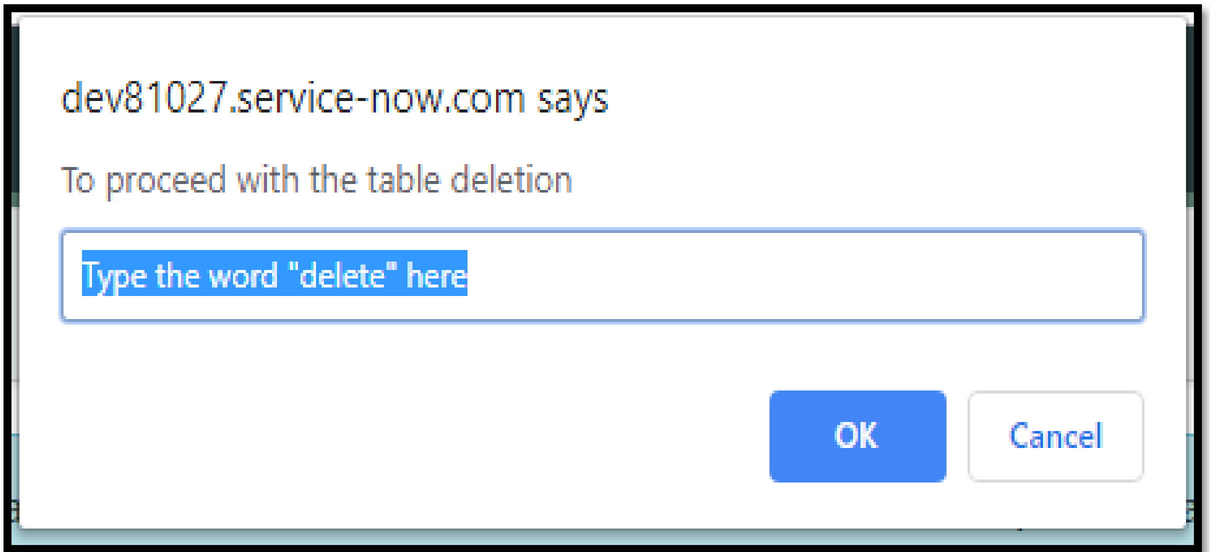
1. Navigate to System **Definition > Tables.**
2. Open any custom table to **delete.**
3. Click **Delete All Records.**



4. In the confirmation dialog box, enter **Delete**
5. And click **OK**.



6. In the confirmation dialog box, enter **Delete**
7. And click **OK**.

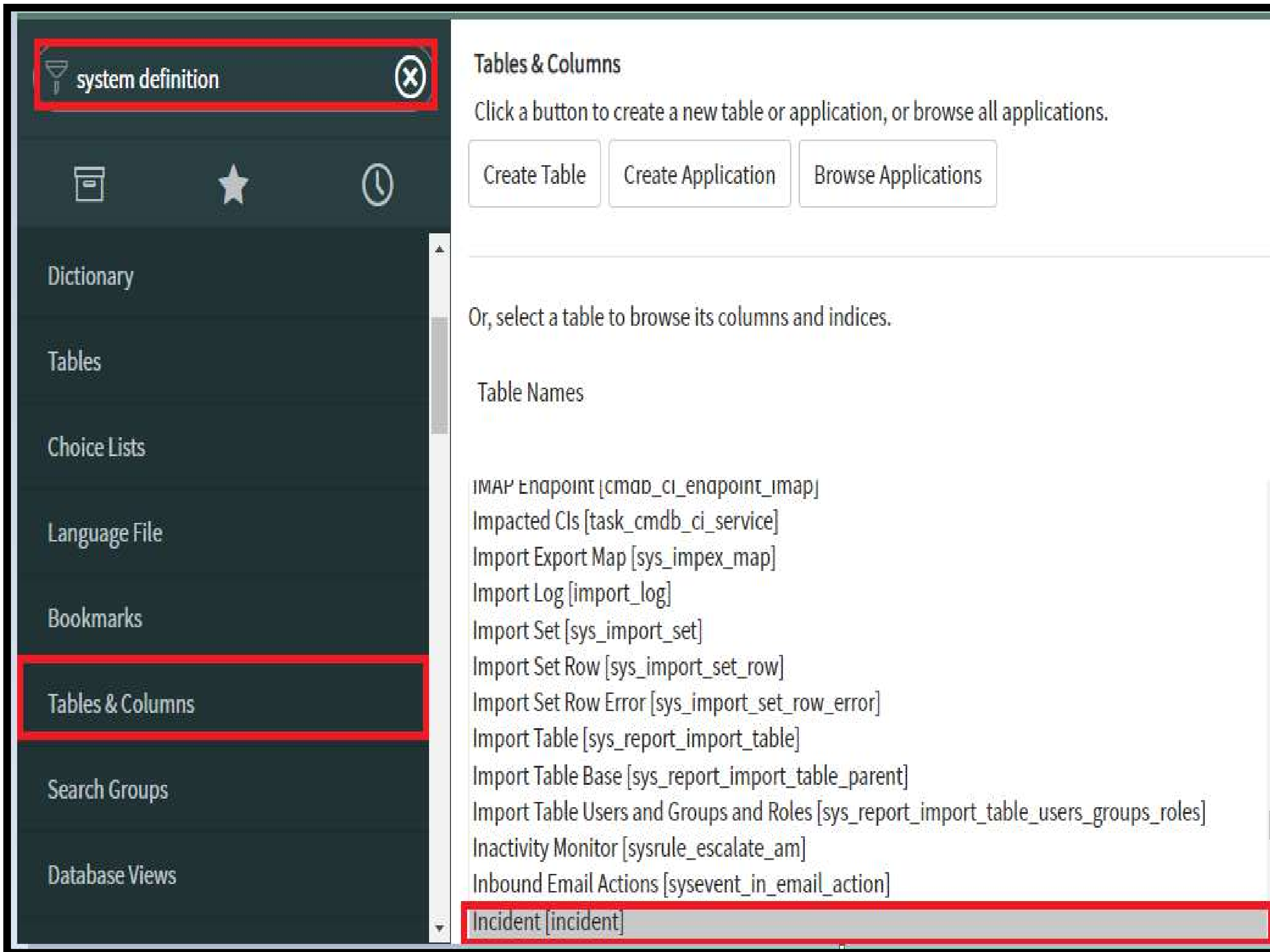


Exercise:3 Delete all records from a table

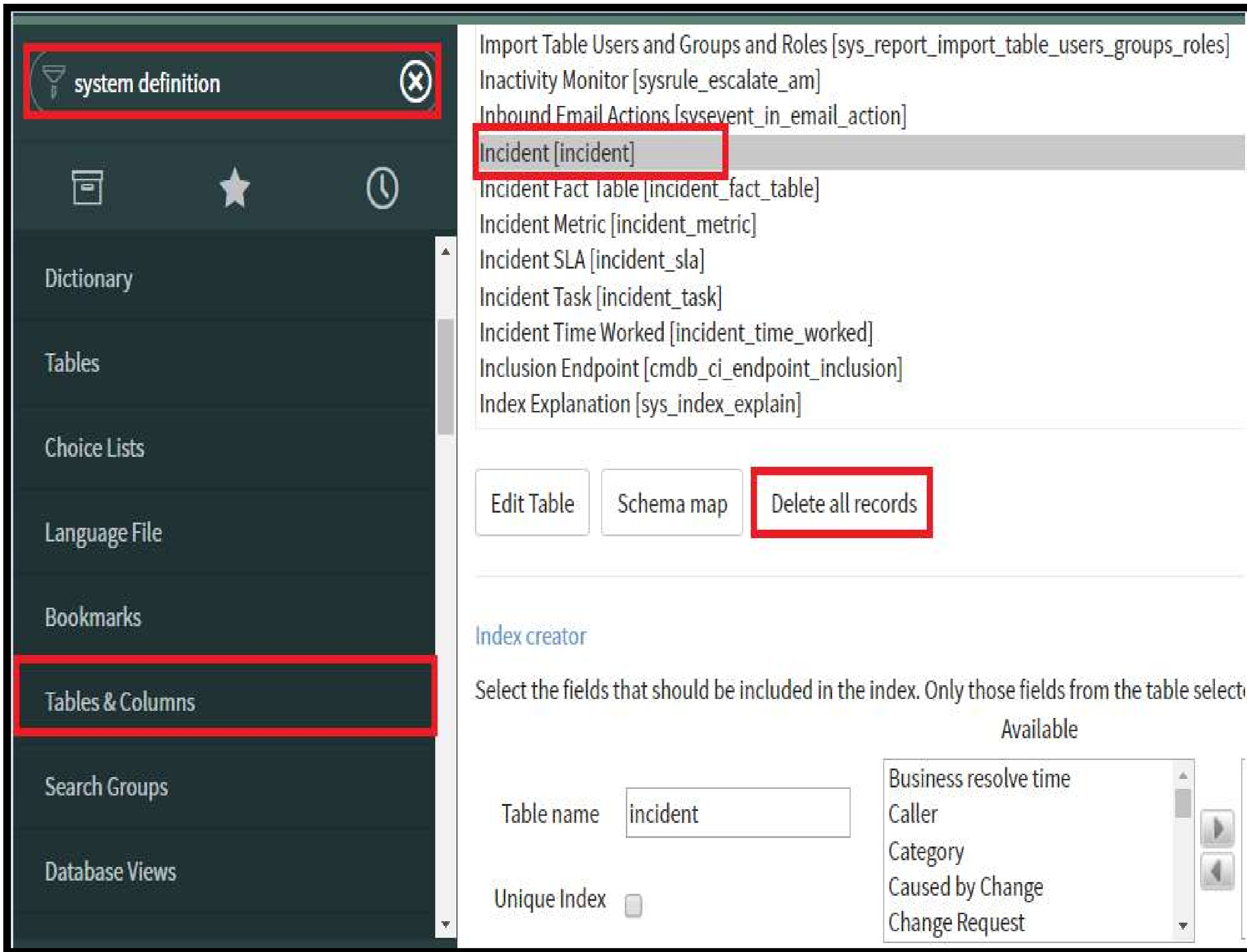
We may decide to delete all the records on a table without deleting the table itself. For example, the administrator may want to delete all incidents on a test instance without deleting the incident table itself.

Procedure

1. Navigate to **System Definition > Tables and Columns.**
2. Select the table for which to delete records.



3. Click **Delete all records.**
4. In the confirmation dialog box, enter **delete and click OK.**



Roll back and delete recovery

Recover deleted records and roll back some actions.

Recover individual records and all related changes, and roll back certain actions such as a patch upgrade, a plugin activation, and background script executions. The recovery and roll backs both use rollback contexts.

The roll back and delete recovery features are available on instances that use **MySQL and MariaDB databases.** Instances that use **Oracle databases only support roll back**. Instances that use **SQLServer do not support roll back or delete recovery.**

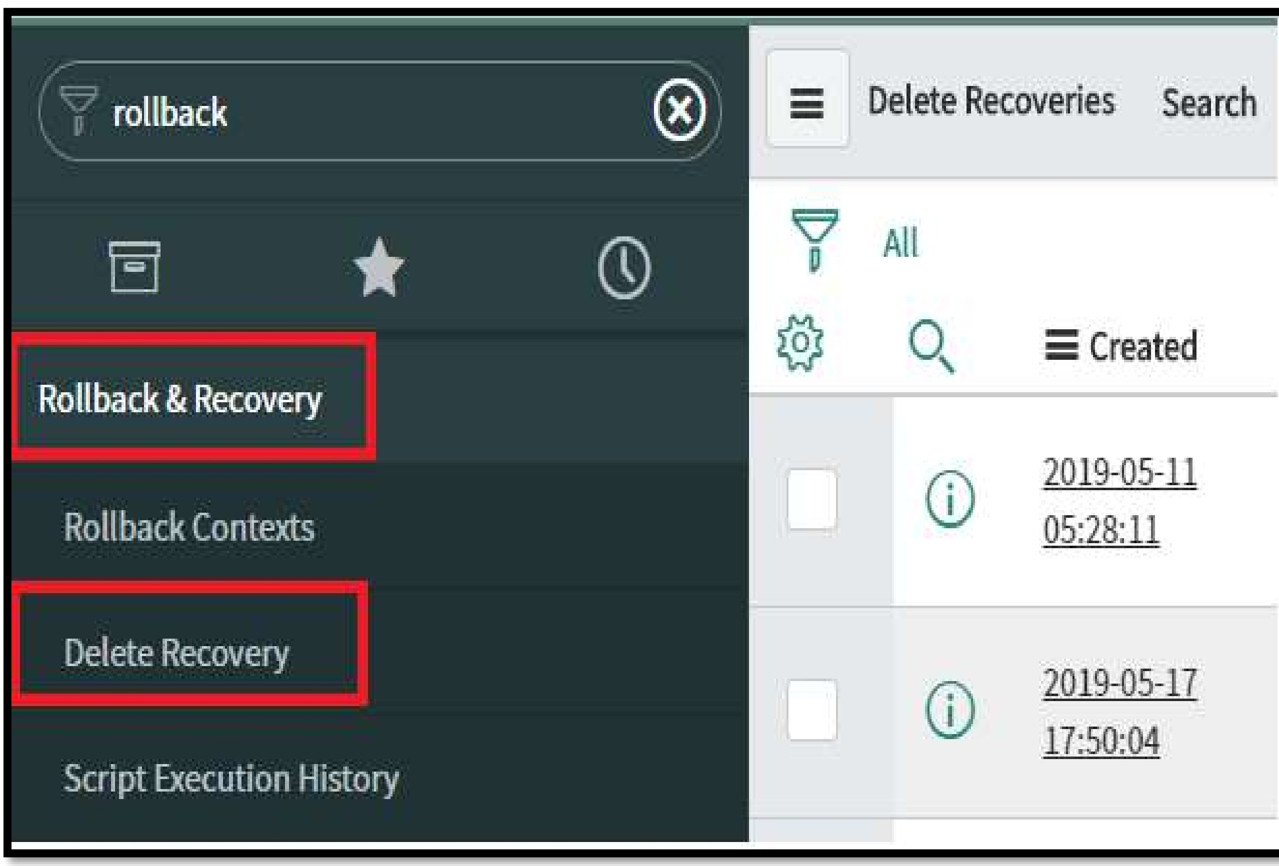
Roll back and delete recovery database support

Database type	Roll back	Delete recovery
MySQL	yes	yes
MariaDB	yes	yes
Oracle	yes	no

Delete Recovery module

This module works for any deleted record. This recovery must be done within **seven days** of the record deletion.

To find this module, navigate to **Rollback & Recovery > Delete Recovery**.



Exercise: 4 restore a deleted record

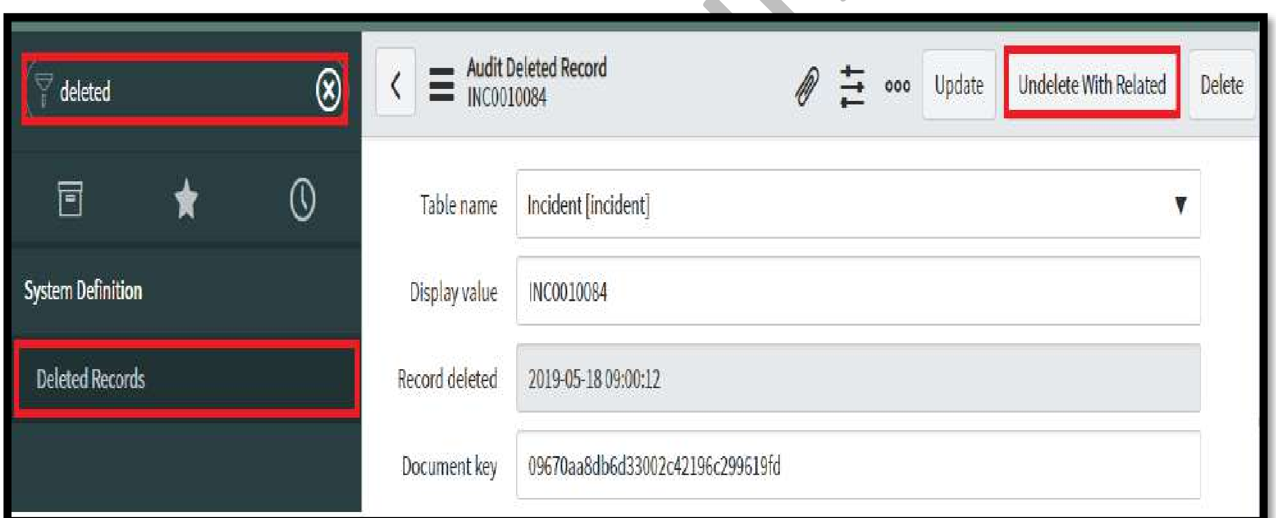
We can recover **deleted records** that are in **audited tables.**

The instance can track deletions on any table, and references on audited tables, with a few exceptions. **Record deletions are not tracked in these circumstances.**

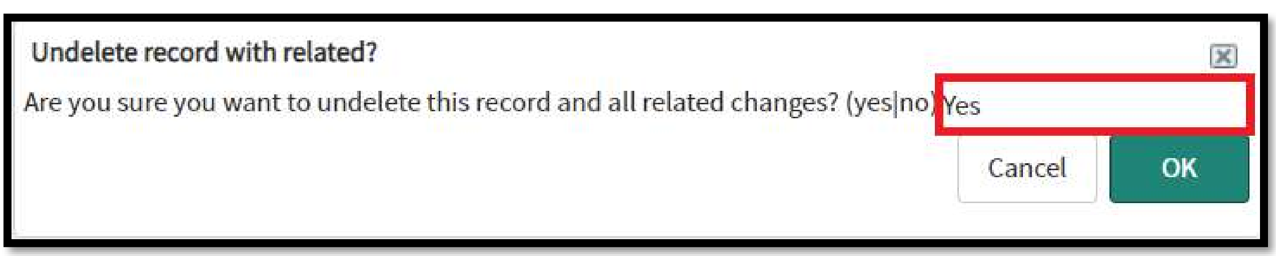
Note : Record deletions are not tracked on tables with the **no_audit_delete=true** dictionary attribute.

Procedure

1. Navigate to **System Definition > Deleted Records.**
2. Open the **deleted record** that you want to restore.
3. **Note:** You can only restore one deleted record and its associated references at a time.
4. Do one of the following to restore the record.
5. Click on **Undelete with related** button



6. In the confirmation dialog box, enter **Yes**



Overview of Task table

Task **[task]** is one of the core tables provided with the base system.

It provides a series of **standard fields** used on each of the tables that extend it, such as the Incident **[incident] and Problem [problem] tables**. In addition, any table which extends task can take advantage of task-specific functionality for **driving tasks**.

The Planned Task plugin provides the Planned Task **[planned_task] table,** which extends the Task [task] Table to provide more fields for tasks to measure duration and effort.

Working with Schema Map

The schema map can define the details of tables and their relationships in a visual manner, allowing administrators to view and easily access different parts of the database schema.

The schema map can also be printed directly from a browser.

Schema relationship types supported, and the colors used for them, are:

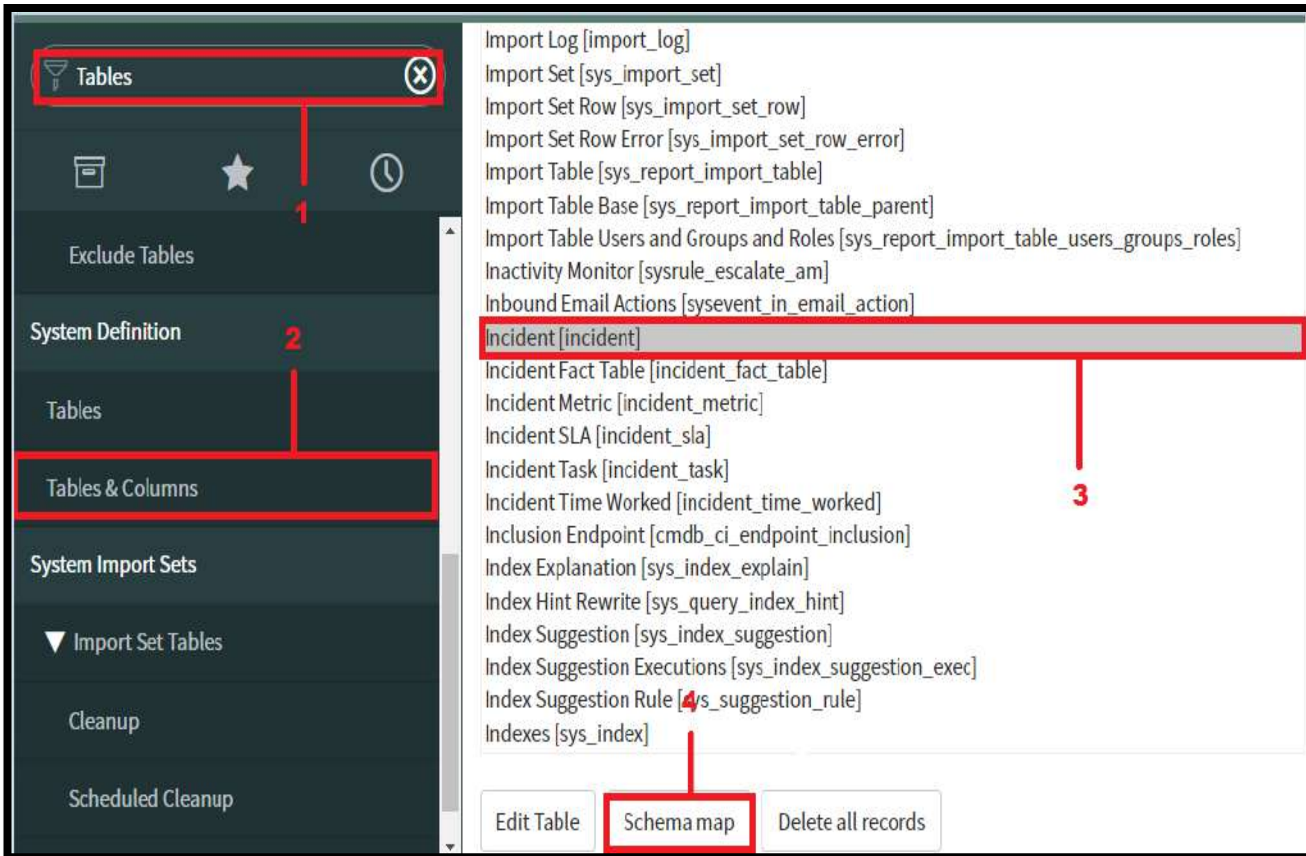
Schema Map for Tables

Schema Map Color Codes	
Referenced by	**Red**
Referencing	**Orange**
Extended by	**Green**
Extending	**Blue**

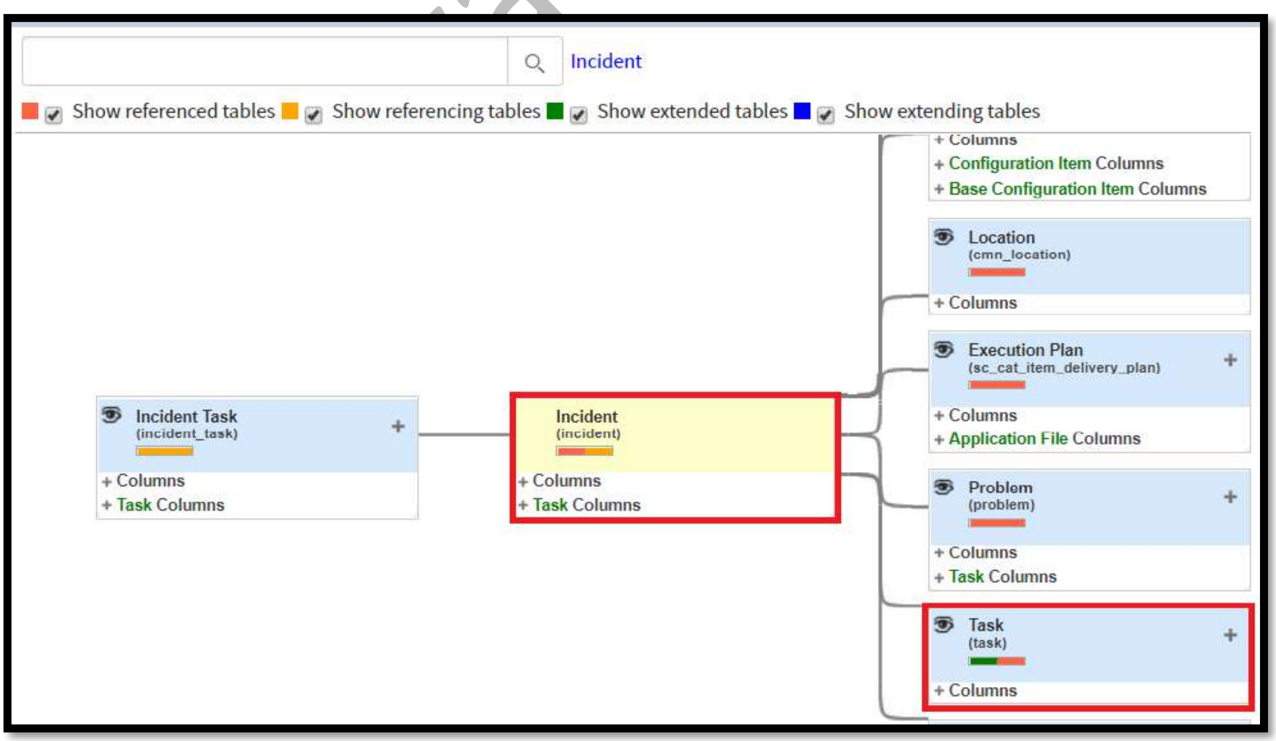
Note: By default, all these types of relationship are displayed, but you can view or hide each type.

Procedure

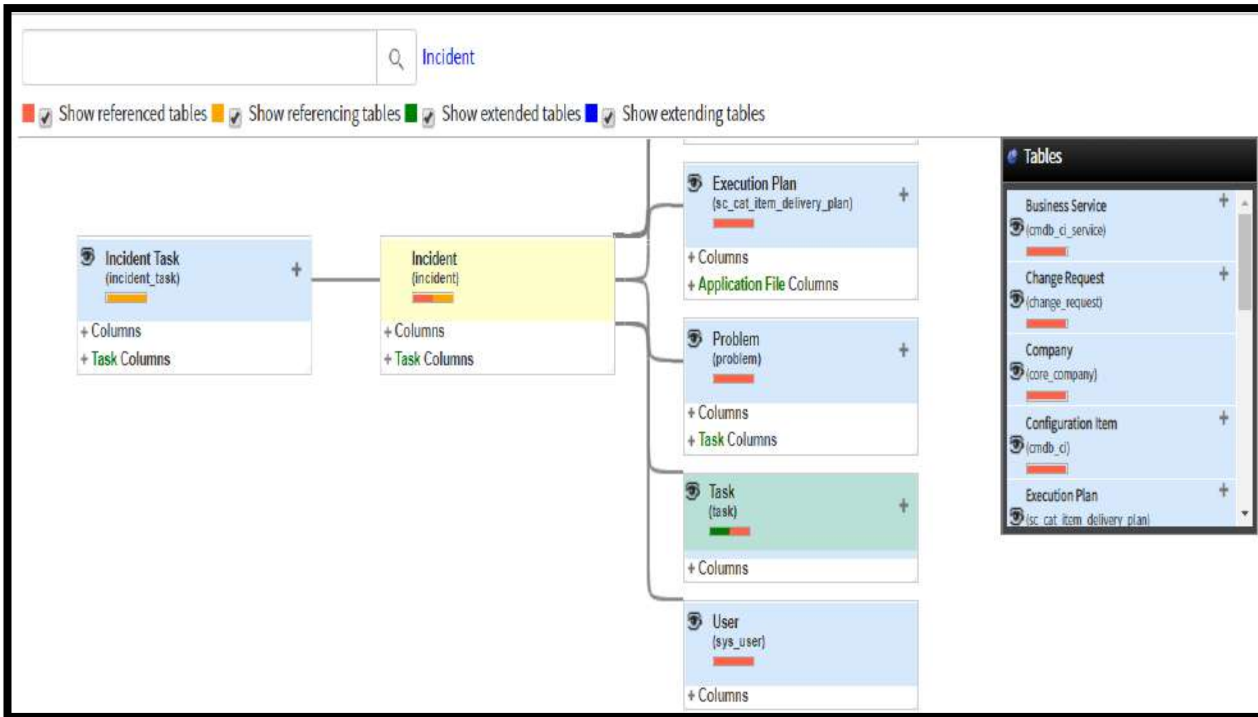
1. Navigate to **System Definition > Tables & Columns**
2. Select any Table like **[Incident]**
3. Click in **Schema Map button**



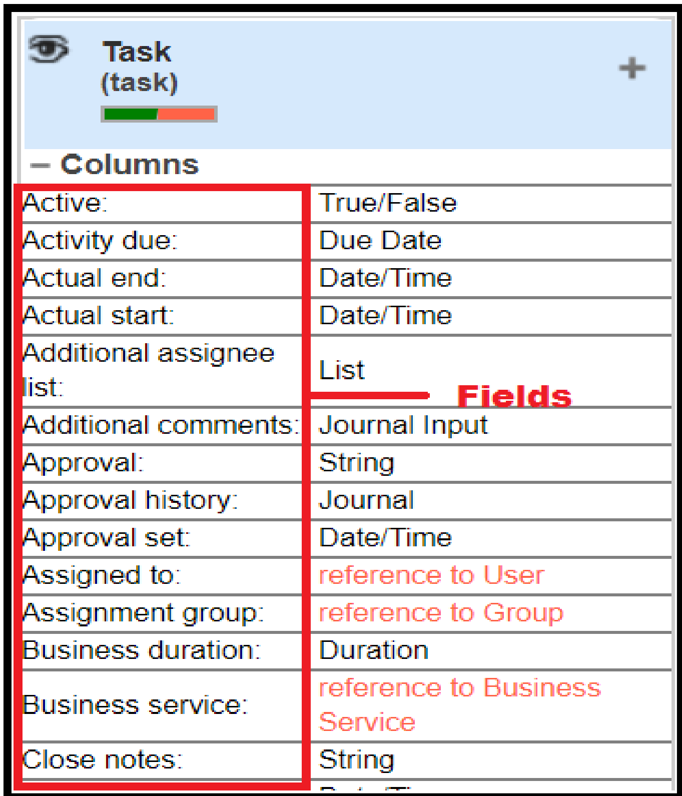
4. Relation can open in **Visualization manner**



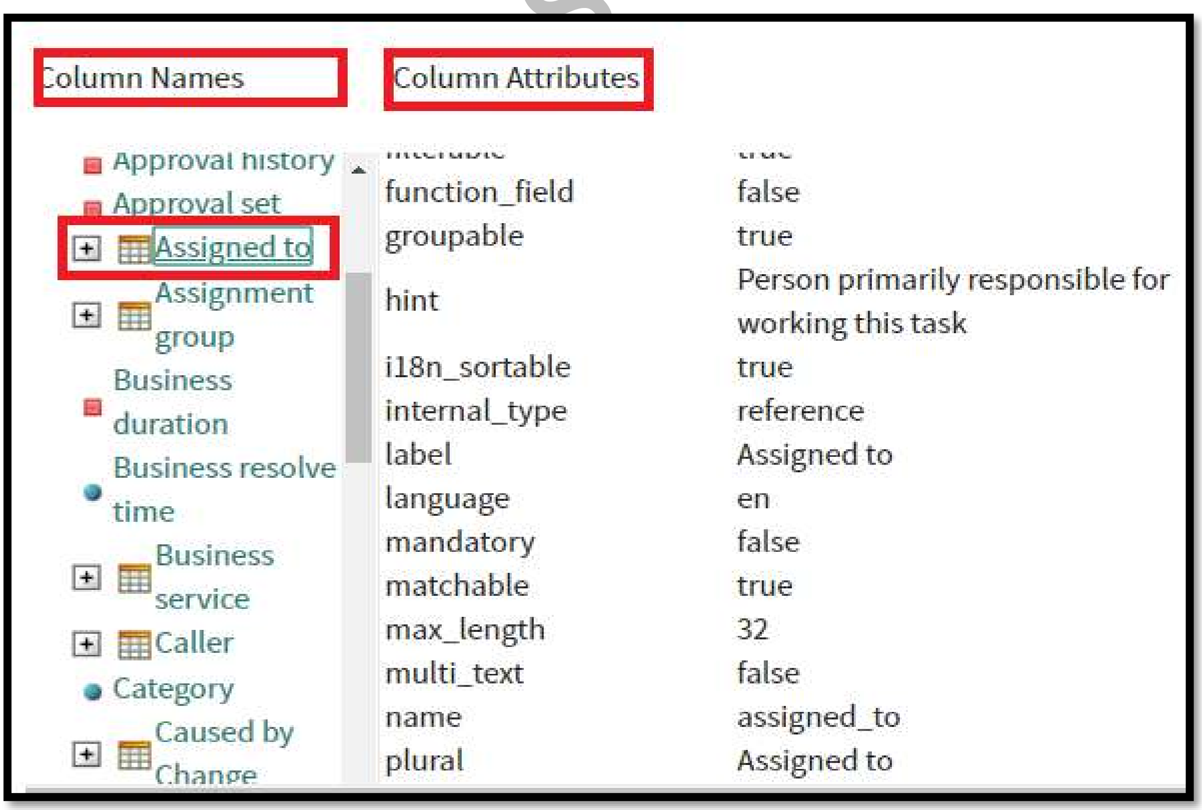
5. The schema map shows the selected table in yellow, typically centered, and all tables related to that table, typically shown at the sides.



6. Each table can display available columns in particular table
7. Click the expand button (**+**) beside **Columns** to expand the table fields.



8. Each field should be maintaining column attributes



Data Dictionary Tables

The system defines data dictionary, data modeling, and entity relationship information in multiple tables.

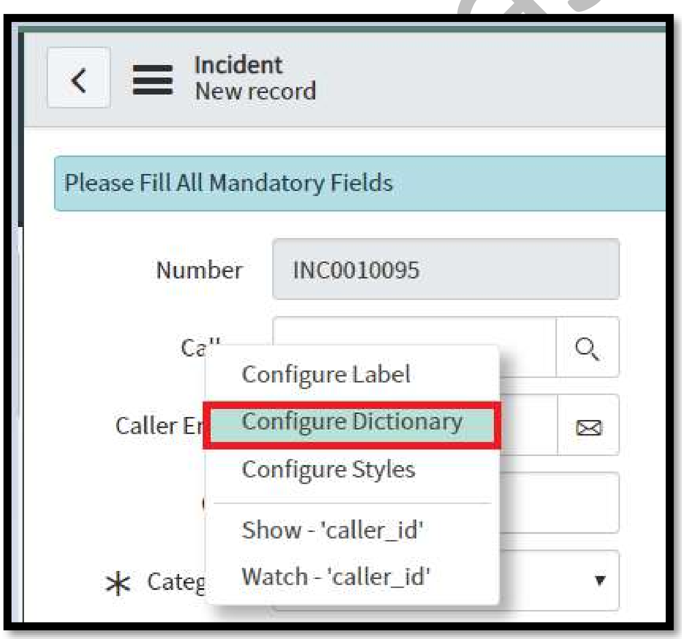
1. Tables **[sys_db_object]:** It will contain a record details for all table.
2. Dictionary Entries **[sys_dictionary]:** It will Contain additional details for each table and the definition for every column on each table. Each row represents either a column on a table or a table.
3. Field Labels **[sys_documentation]:** It will contain the labels and language information.

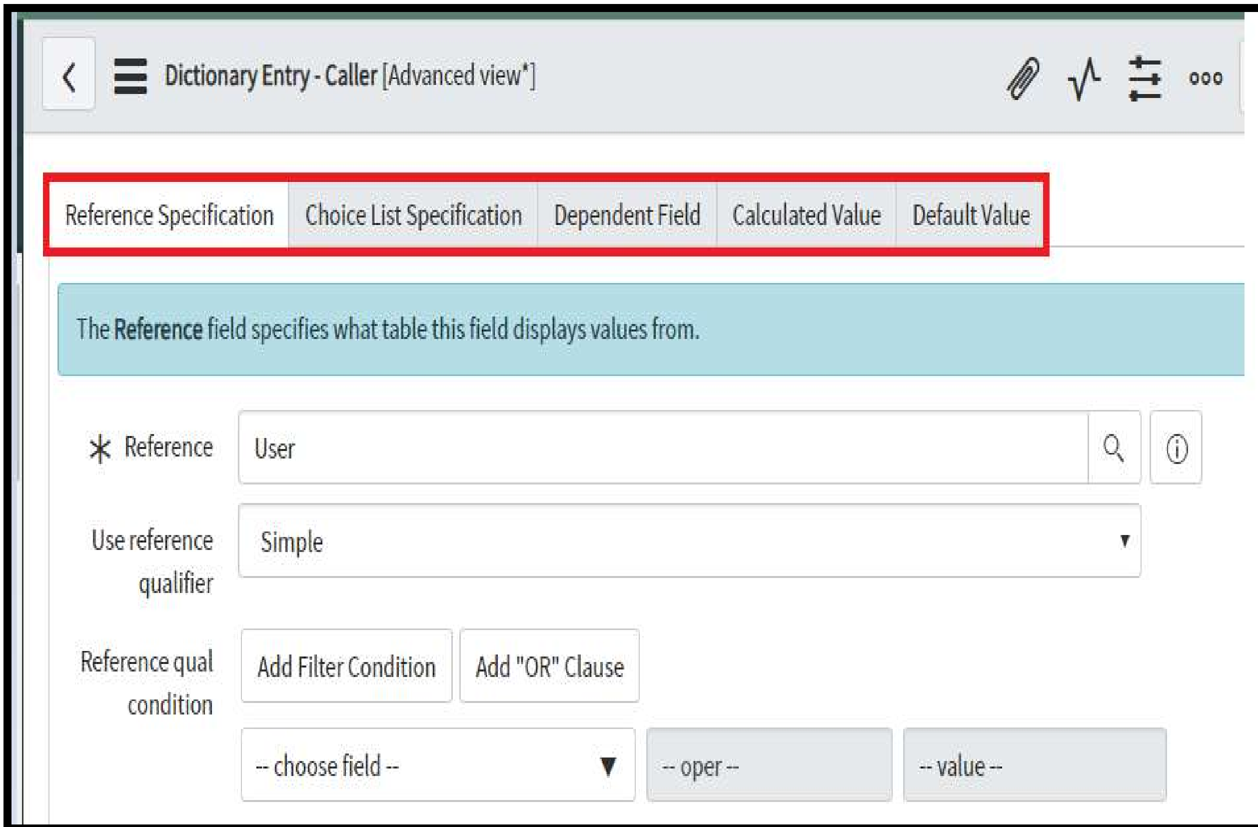
Dictionary Entries

The Dictionary Entries **[sys_dictionary] table**, also we called the System Dictionary, it will contain every table and field details in the system. It will contain information about fields **data type and character limit, setting a default value, dependency field setup, and other attributes of a field.**

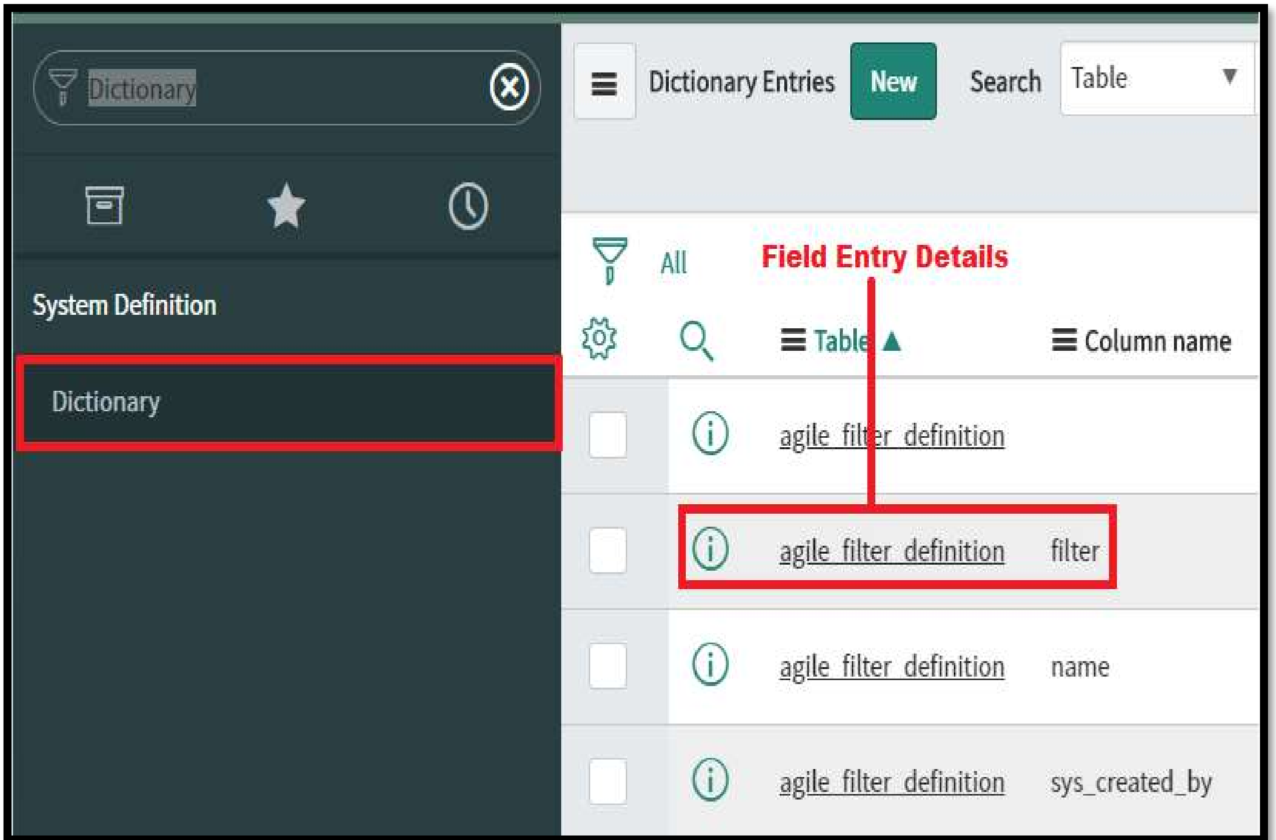
We access the system dictionary tables in **multiple ways.**

1. Navigate to **System Definition** > **Dictionary**.
2. To view particular dictionary definition, right-click the list header, form header, or field label, and select **Configure Dictionary**.
3. Open **incident** form and right-click on **Caller Label** then see following screen shot below





4. Second way Navigate to **System Definition > Dictionary**

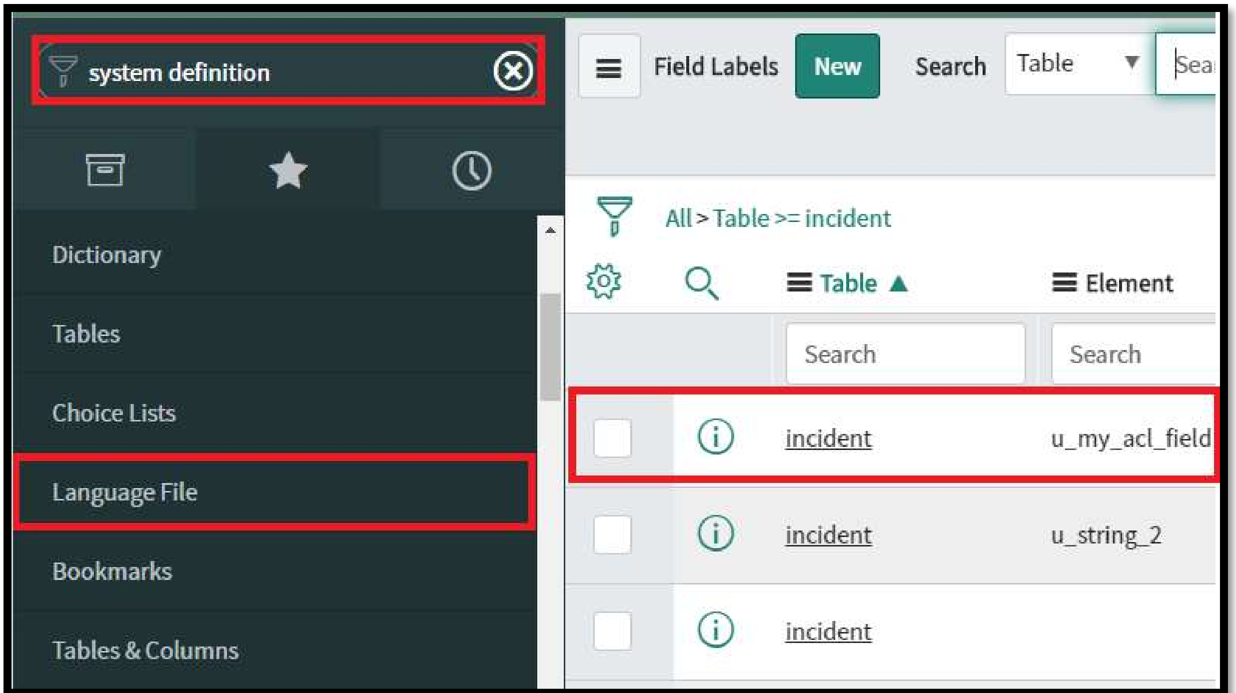


Field Labels

The Field Labels **[sys_documentation]** table, also called the Language File, contains information about each field labels and hints for each table and column in the system.

Access the language file in one of these ways:

1. Navigate to **System Definition > Language File.**
2. To see the field label for a particular field, right-click on the any field label on the form.
3. The following image shown the language file filtered on Incident table



Dictionary overrides

Dictionary overrides can provide the details about field on an extended table differently from the field on the parent table.

For example, for a field on the Task **[task] table**, a dictionary override can change the default value on the Incident [incident] table without affecting the default value on **Task [task] or on Change [change].**

Only administrators can override these aspects of a field:

- ✓ Reference qualifiers
- ✓ Dictionary attributes
- ✓ Default values
- ✓ Calculations
- ✓ Field dependencies
- ✓ Default column display values
- ✓ Mandatory and read-only status and display

Use a dictionary override to allow a field in a **child table** to have a different value or behavior than the same field in a **parent table.** For example, a dictionary override changes the default value of the priority field from **3** in the parent table to **5** in the Incident table. **child table**

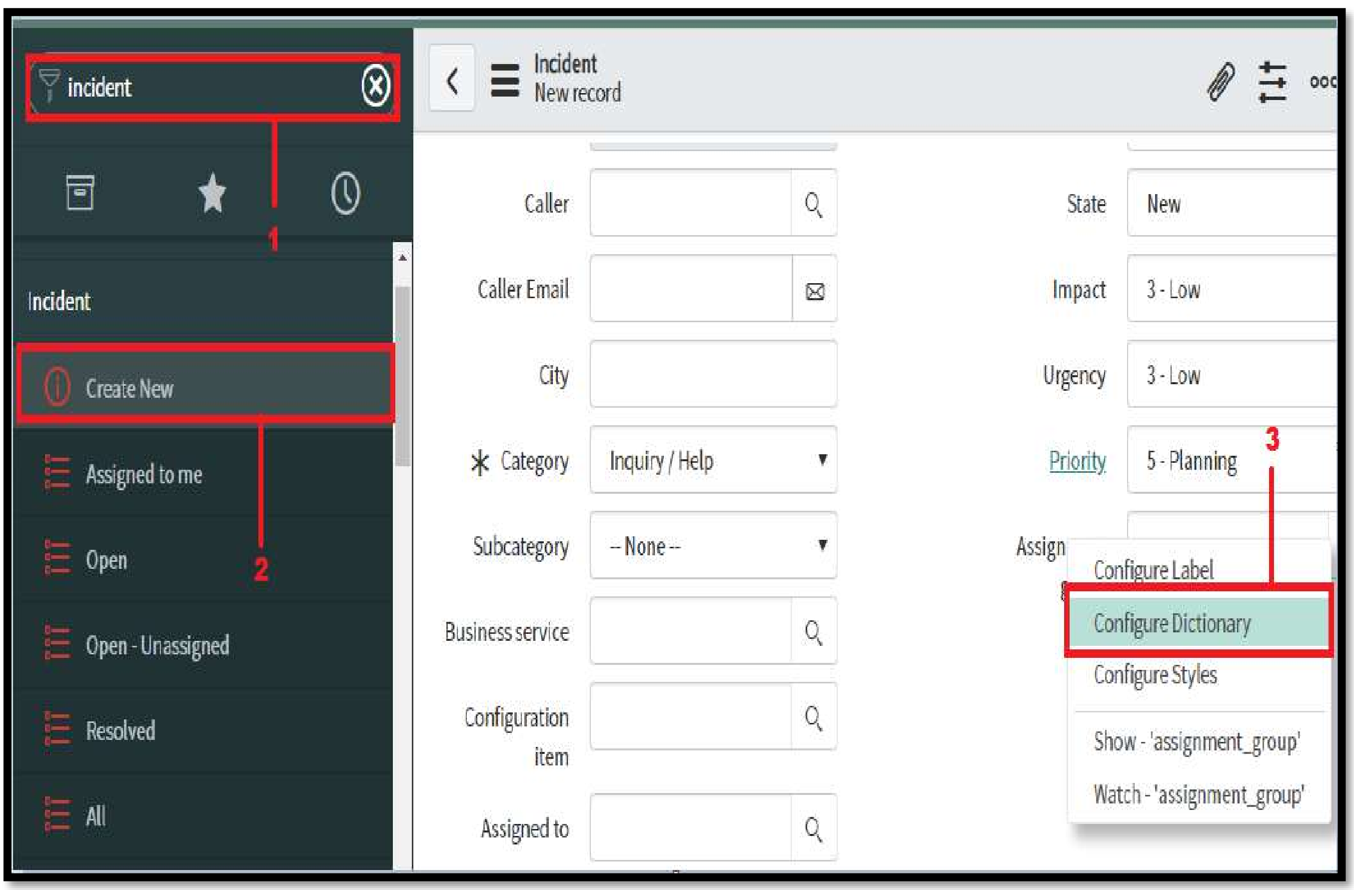
Exercise: 5 Configure dictionary Override

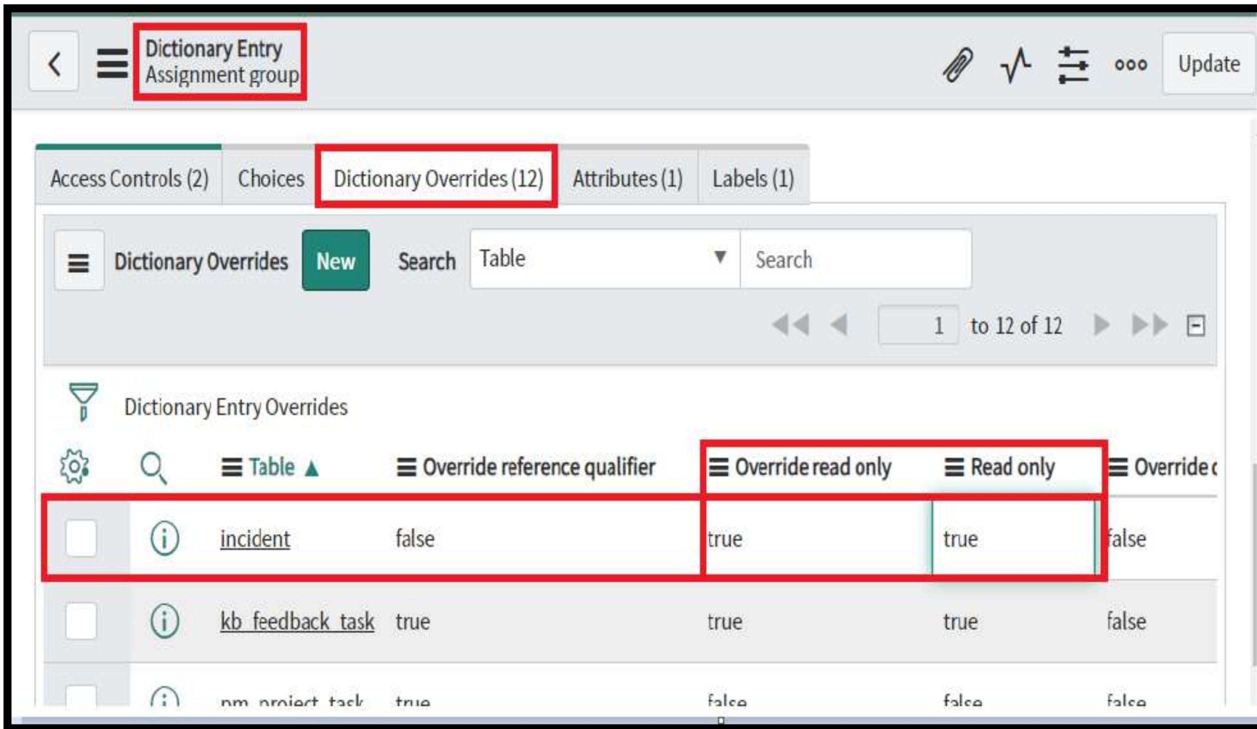
Dictionary override used to override fields default behavior

Procedure

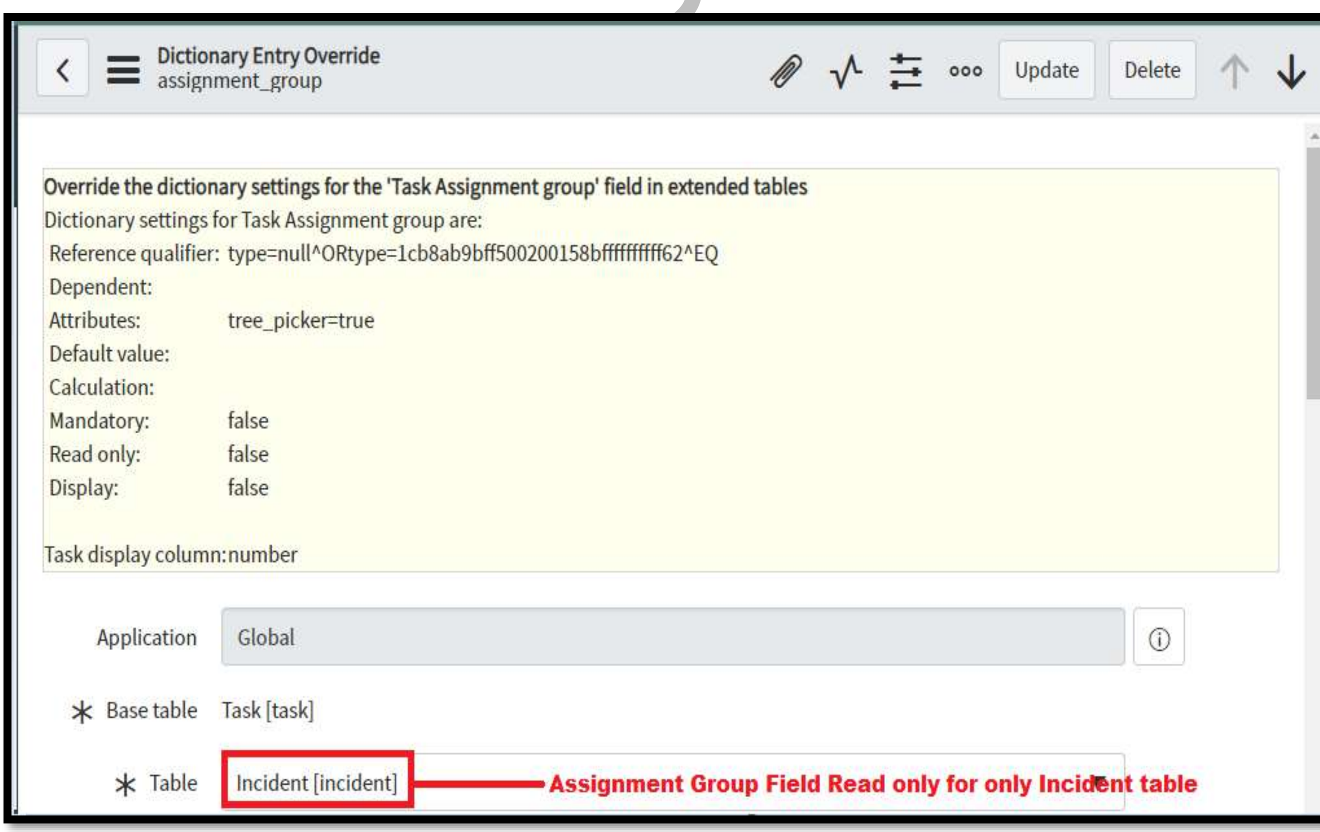
Task: Assignment_group field is read only for only **incident** table

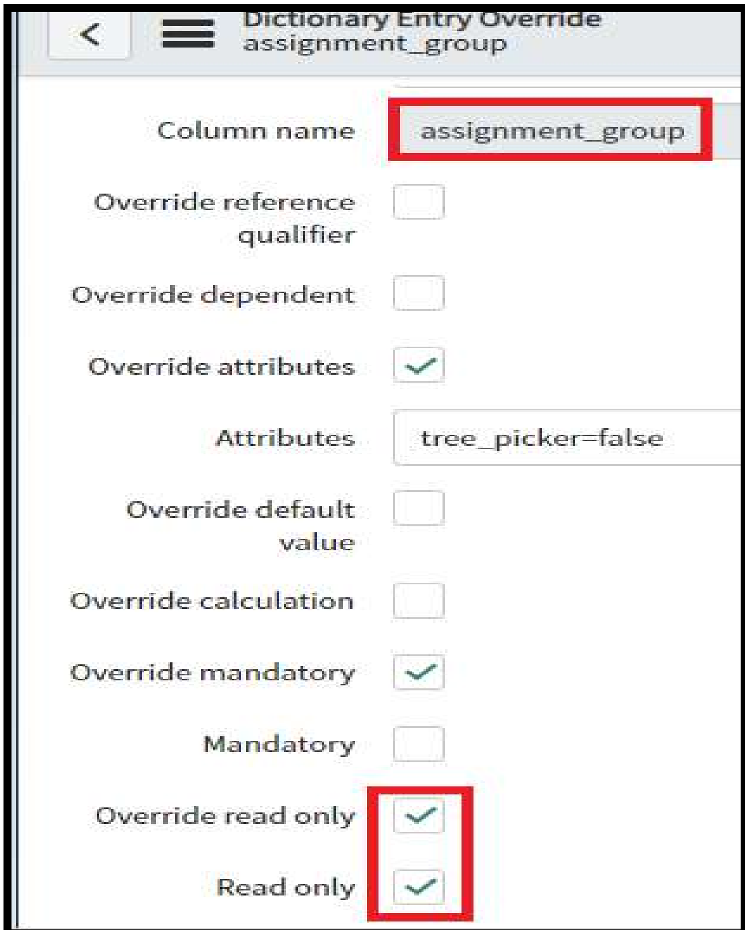
1. Navigate to **Incident Table**
2. Click on **New** module
3. Right click on **assignment_group** field
4. Click on **Configuration Dictionary**
5. In the **Dictionary Overrides** related list,
6. Click on **New**.
7. Fill in the details on the form





8. Make **assignment_group** field is read only for **Incident** table
9. Open the **Record**





10. Click on **Update**

Chapter – 8

User Administration

User Administrator Overview

Manage the individuals who can access your instance by defining them as users in the system. Create user groups, and assign users to them. Use roles to specify what different users and user groups can see and do.

A group is a **set of users** who share a common purpose. Members of groups perform tasks such as approving change requests, resolving incidents, receiving email notifications, or performing change request tasks. Assign every user to at least one group.

Best practice of assigning roles

A good practice is to assign roles to **groups rather than to users**. Administrators can assign roles quickly to multiple users by adding the users to the group. If a user moves to another group, the role assigned to the new group is applied automatically.

1. Users are a critical part of any organization, and application access should be smooth,
2. for streamlined access to applications, you can integrate Service-Now with your company's
3. LDAP server. Out of the box, Service-Now provides an LDAP server and user
4. Administration modules for configuration.

Allow a user to view own profile

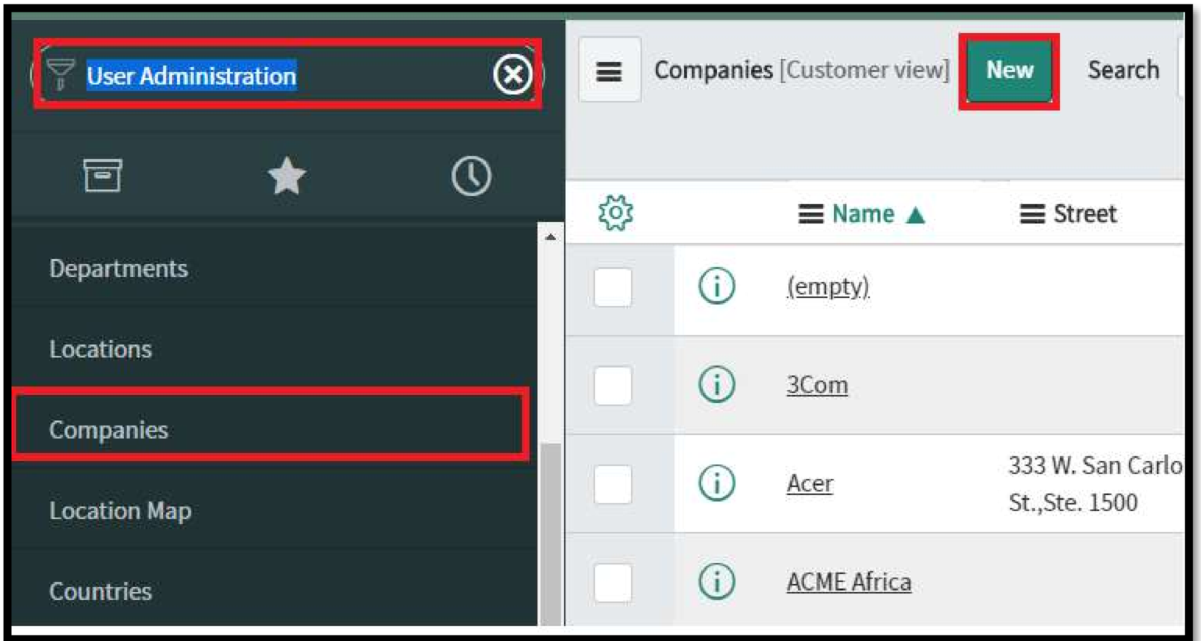
Current logged in Users are always able to view their own profile by clicking their name in the Welcome banner. In case if your users cannot do this, enable a system property.

Exercise:1 How to add a new company

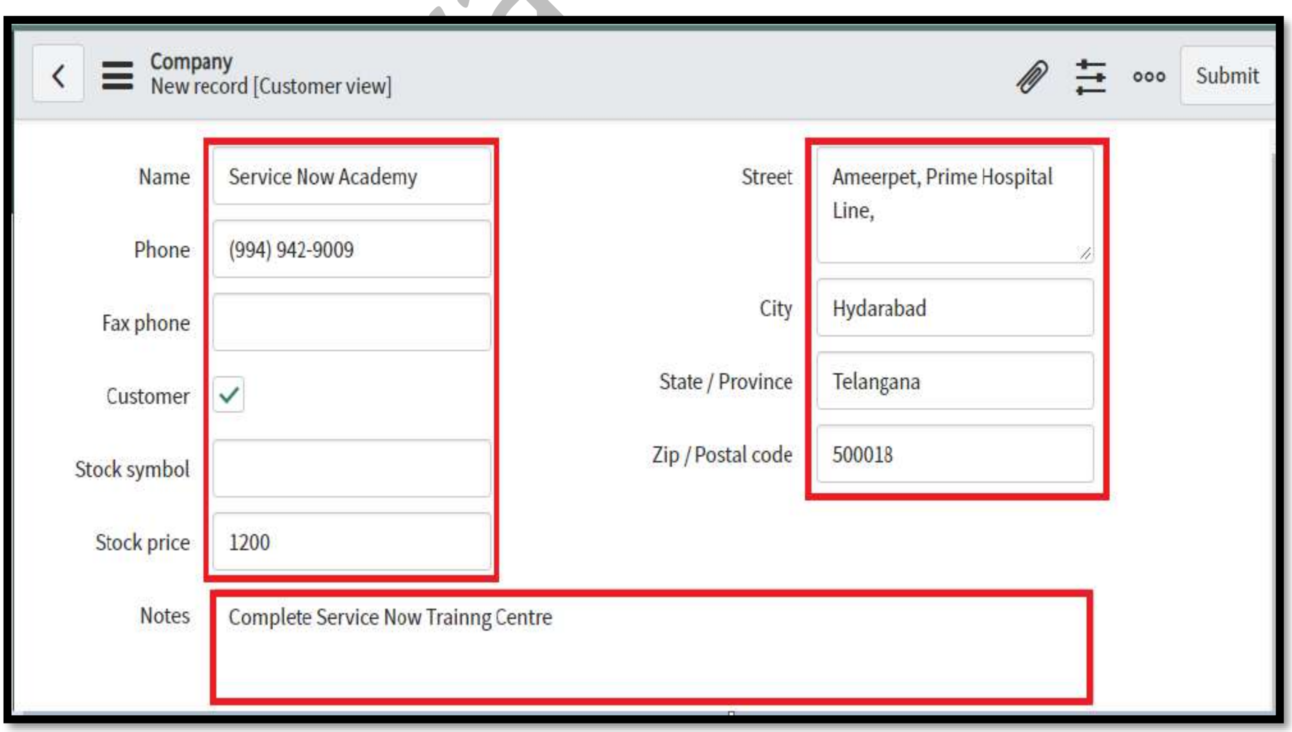
We can add new companies that represent **third-party vendors, manufacturers, or customers** with whom you do business. These companies provide a way to categorize **users, groups, and assets.**

Procedure

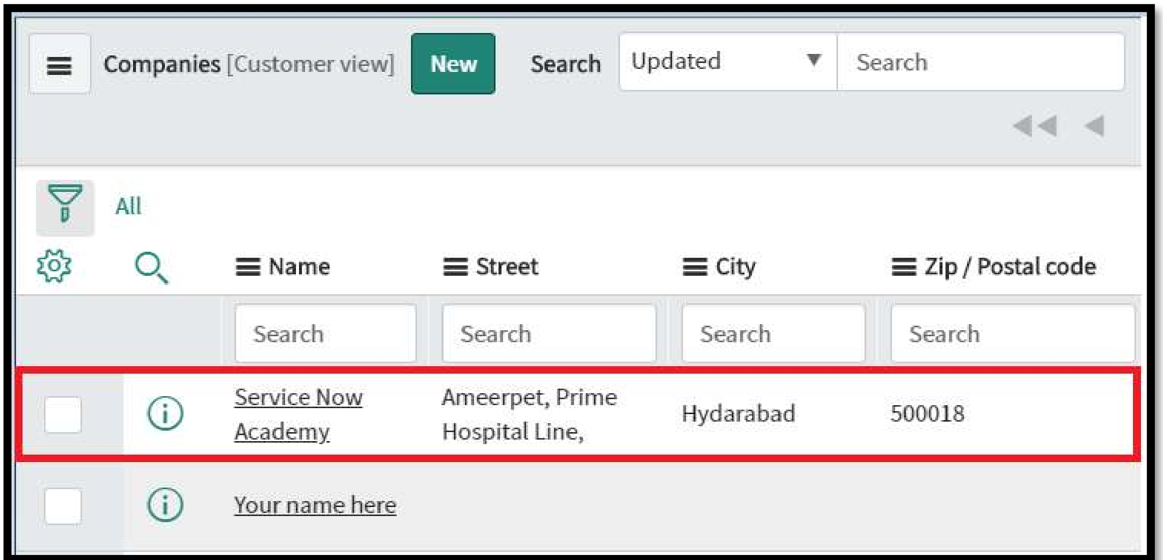
1. Navigate to **User Administration > Companies.**
2. Click **New**.



3. Fill and complete the form with details



4. Click on **Submit**
5. Check company record in Companies **list View**



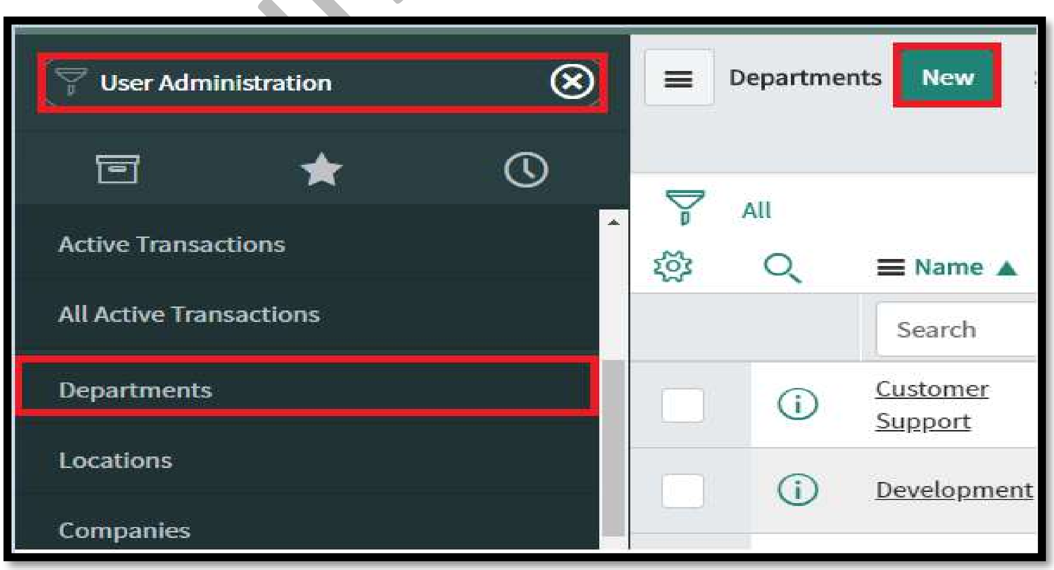
Exercise: 2 How to add a new department

Departments provide another way to categorize **users, groups, and assets**. You can add departments and assign them to **users and groups.**

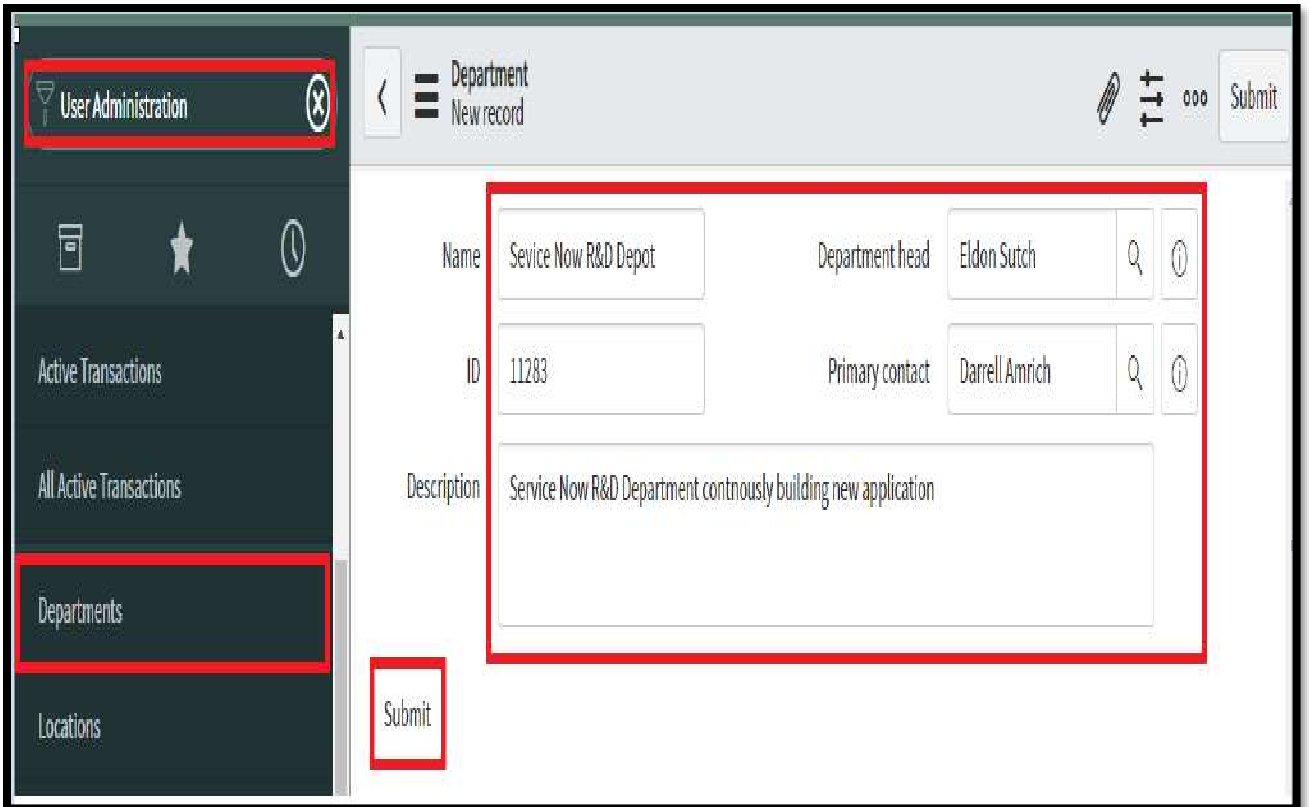
Department records are stored in **common_department(**table)

Procedure

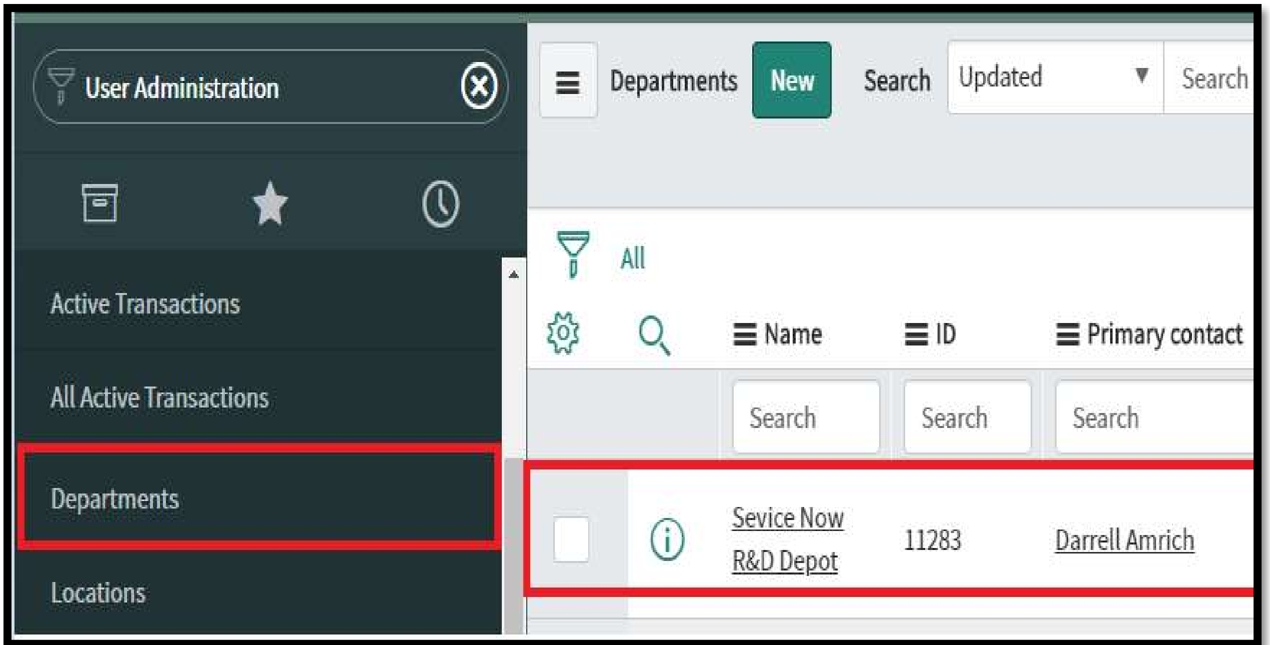
1. Navigate to User **Administration > Departments** and
2. Click on **New** record.



3. Create or modify the **department name, ID, and description.**
4. (Optional) Select the company the department is associated with.
5. (Optional) Add a **department head, primary contact**, or both from your list of users.
6. (Optional) Add the parent department, if applicable.



7. Click **Submit.**
8. Check in **cmn_department. list** in list view



Working with Users

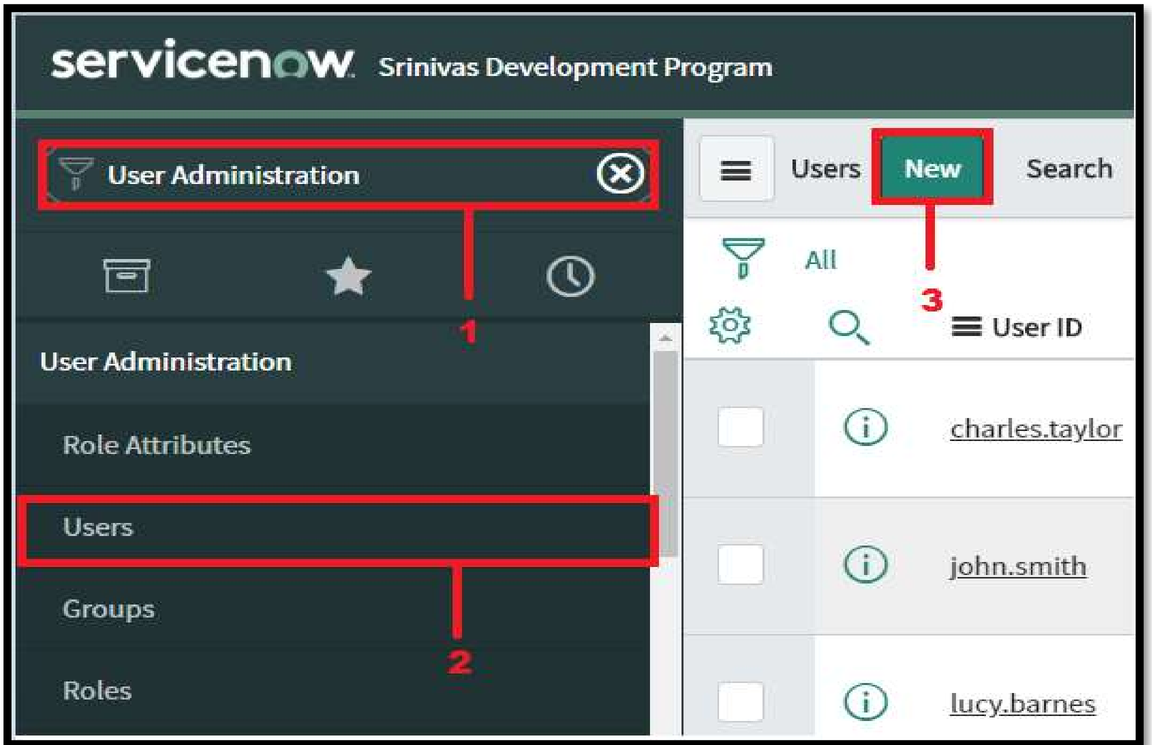
We can add a user to your instance to allow them to log in and use the service now application features.
User records are stored in **(sys_user)** Table

Exercise: 3 How to create a new User

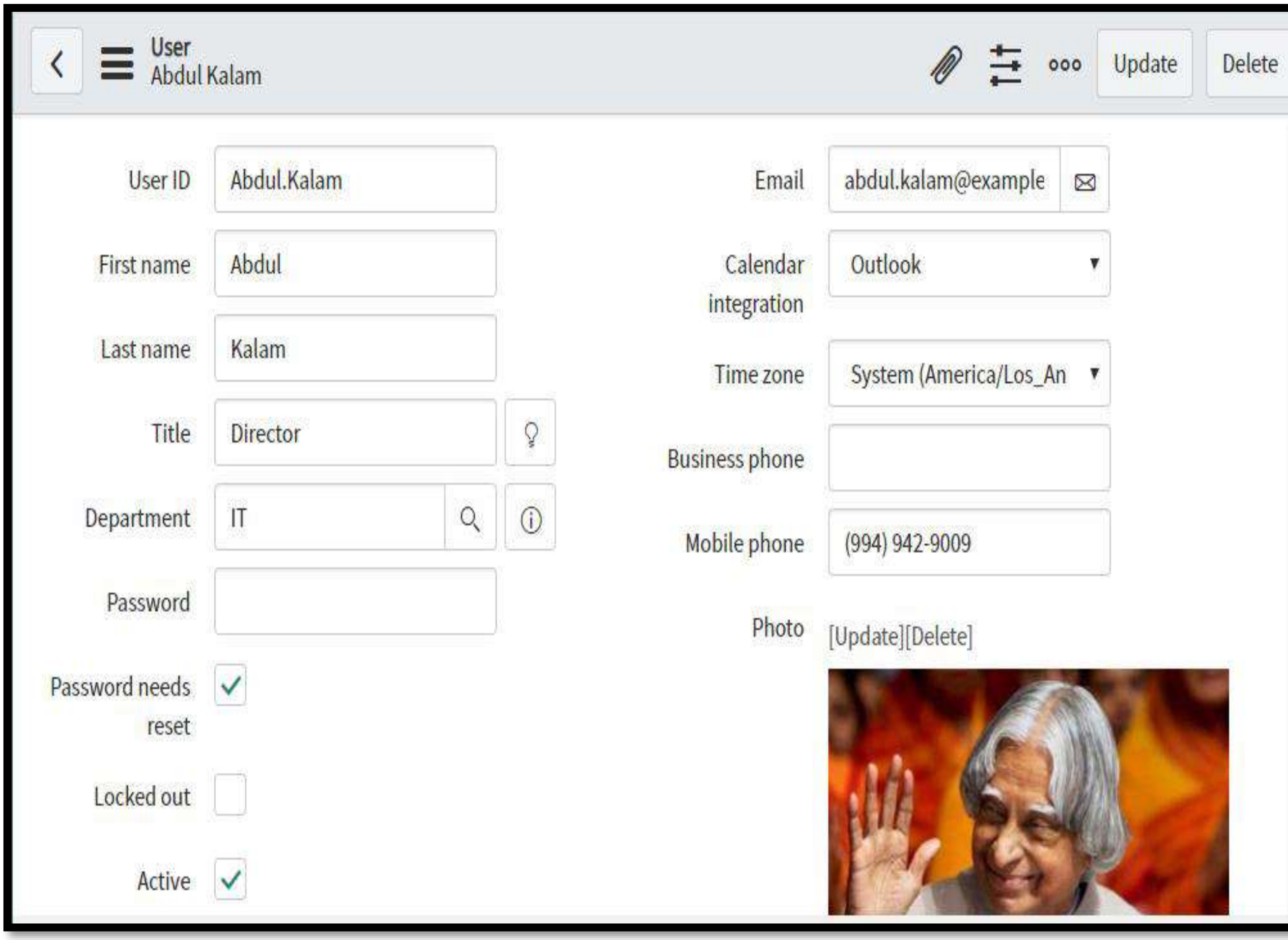
Service now Administrator can create new users in service now instance to access objects and modules in system

Procedure

1. Navigate to **User Administration > Users.**
2. Click **New** and enter the following information:



3. Fill the required details below
4. **Submit**



Password needs reset: Enable the check box to require the user to change the password during the first login time.

Locked out: Enable the check box to lock the user out of the instance and terminate all their active sessions. The system prevents users with the admin role from locking themselves out.

Active: Enable the check box to make this user active.

Note: Only the administrator sees inactive user in:

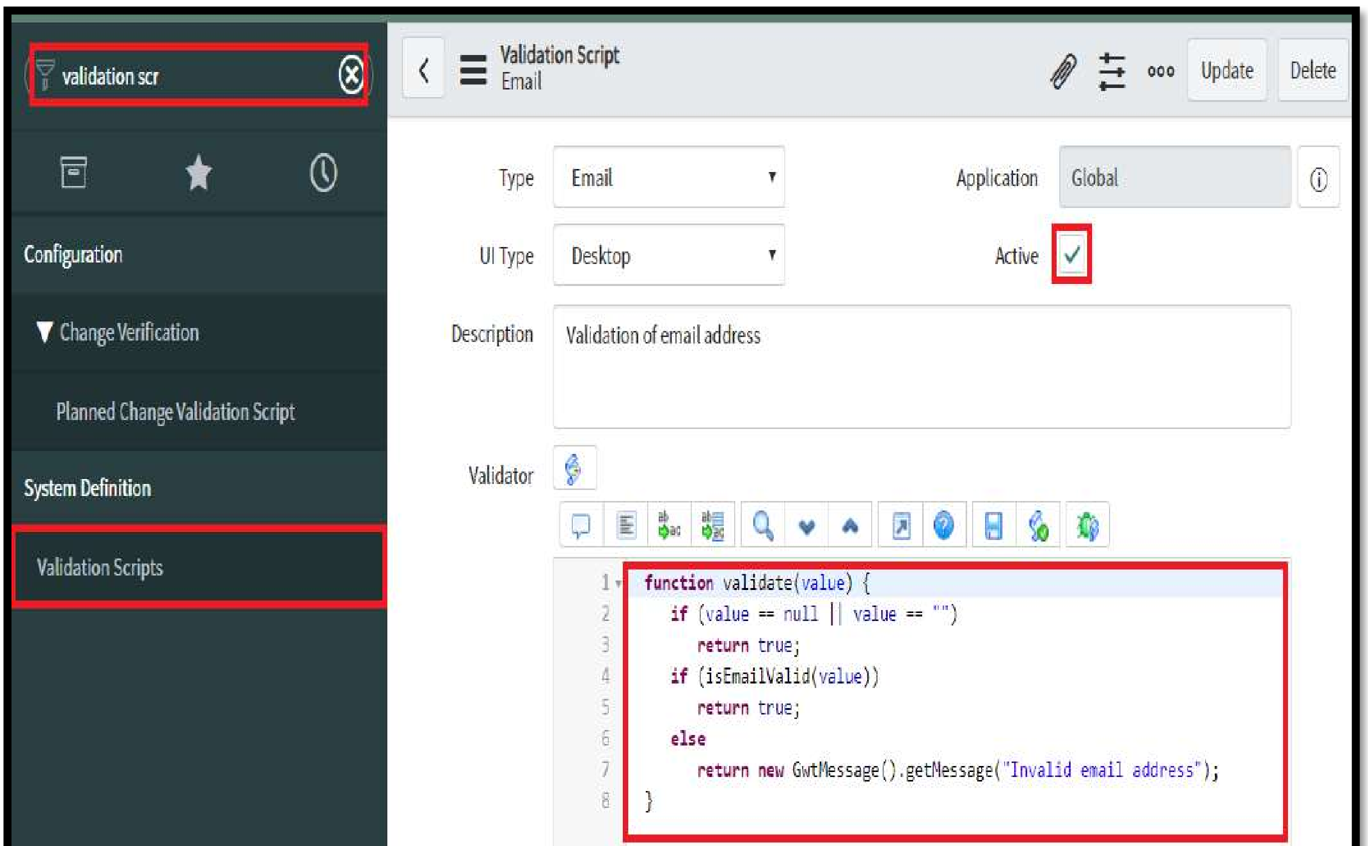
Date Format: Select the **user's** preferred format for dates.

Exercise: 4 User Record Email Validation

Any user they want enter a non-standard email address in email field that does not pass field validation, you must deactivate the validation script first. then provide non-standard email id.

Procedure

1. Navigate to **System Definition > Validation Scripts**.
2. Select the **email** record.
3. uncheck the **Active** check box and save the change.
4. Complete the user profile, including the **email address**, and update or submit the record.
5. **Enable** the email validation script check box.



6. Above screen shot will define how email validation working

Impersonate a user

Only Service Now Administrators can impersonate other authenticated users for testing purposes and view impersonation logs. **The impersonation option is not visible in the mobile view** of the platform.

When we impersonating another user, the administrator has access to exactly what that user can access in the system, including the same **menus and modules**. The instance records anything the administrator does while impersonating another user as having been done by that user.

Impersonation Specifications

If we want user account to be impersonated should have a **user ID**. We can find this ID in the User **[sys_user]** table record for the account.

Note: If The use record is missing or not available, then given message to the user you selected could not be impersonated appears.

We need several different logins to test the system:

1. An **admin** account to do work.
2. An **itil** (or similar) login to test as a technician.
3. An **ess (Employee Self Service)** login to test as an end user desktop.

Impersonation logs

Impersonations are logged in the System Log.
Log impersonations for either interactive (UI) or non-interactive sessions.

Working with Groups

A group is a set of users or customers who share information for common purpose.

Groups may perform tasks such as approving change requests, resolving incidents, receiving email notifications, or performing work order tasks. Any business rules, assignment rules, system roles, or attributes that refer to the group apply to all group members automatically. Users with the **user_admin** role can create and edit groups.

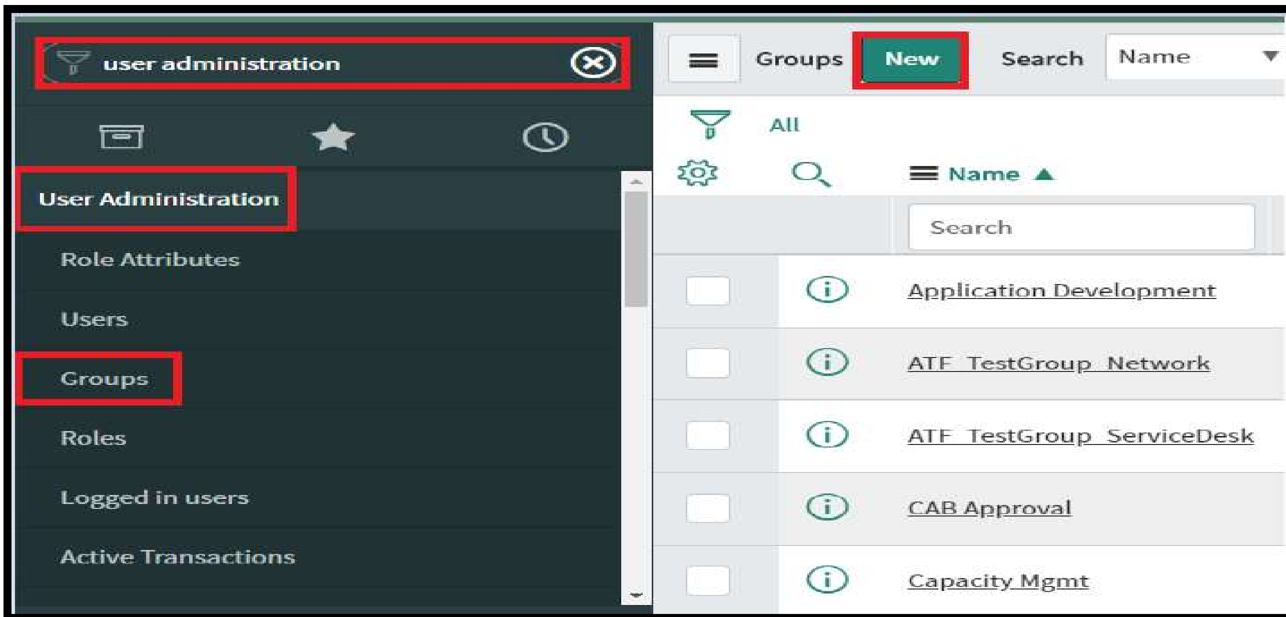
1. A record in the **sys_user_group** table
2. Buckets which hold users who share a common purpose
3. Roles are assigned to groups
4. Groups contain **0 or more** roles
5. One Inherit other groups

Exercise: 5 How to create a new group

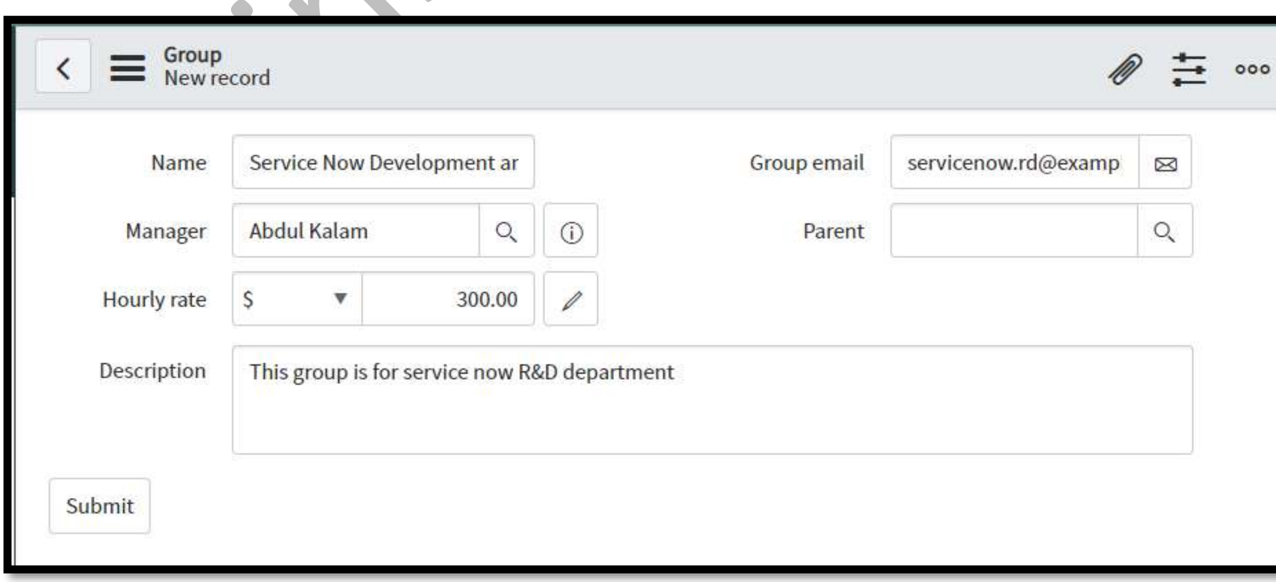
We can create groups and assign roles to them. Users assigned to the group inherit the roles.

Procedure

1. Navigate to **User Administration** > **Groups**
2. and create a **New** record
3. Fill the form with required details



4. Click on **Submit**



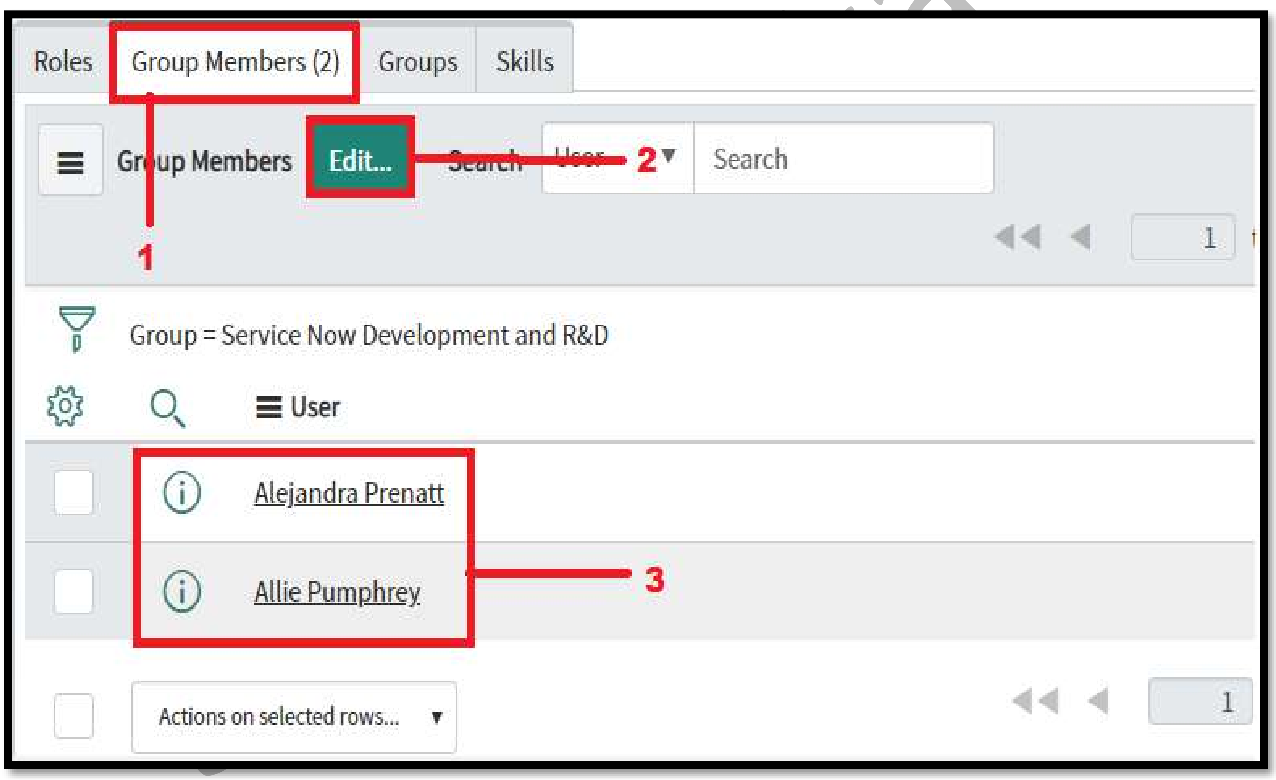
Exercise: 6 Add a user to a group

We can add a user or members to a group so the user inherits all the roles assigned to the group.

Note: Only Admin can add members to specific group

Procedure

1. Navigate to **User Administration > Groups.**
2. Open particular group record.
3. In the **Group Members related list**, click **Edit**.
4. Select one or more names in the Collection list.
5. Click **Add.**
6. Click **Update** the record.



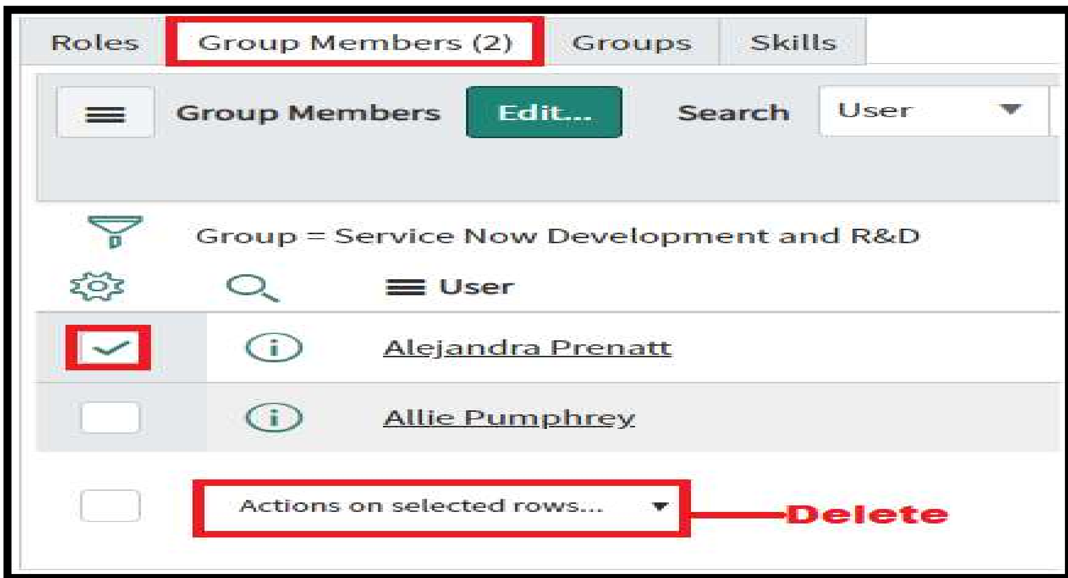
Exercise: 7 Remove a user from a group

We can remove a user from a group when they change roles.

Note: Before selecting **Delete**, first make sure you have properly selected the rows containing the specific users you wish to remove from the group **group**.

Procedure

1. Navigate to **User Administration > Groups**.
2. Open particular group record.
3. In the **Group Members related list**, select the check box next to each group member name you wish to remove.
4. From the Actions on selected rows menu, select **Delete.**



Working with role

A role is to control access to features and functionality in applications and modules. The new role does not have access to any application or module until you add other roles to it or add the new role to the appropriate applications and modules.

1. Grants permissions to parts of system
2. A record in the **sys_user_role** table
3. Assign **1 or more roles** to groups
4. Role contain 1 or more access control rules
5. Many OOB roles Service Now platform supports the following types of users:
6. End User
7. IT User **(ITIL or itil_admin)**
8. Approver (**sys_approver**)
9. System Administrator **(admin)**

Base System Roles

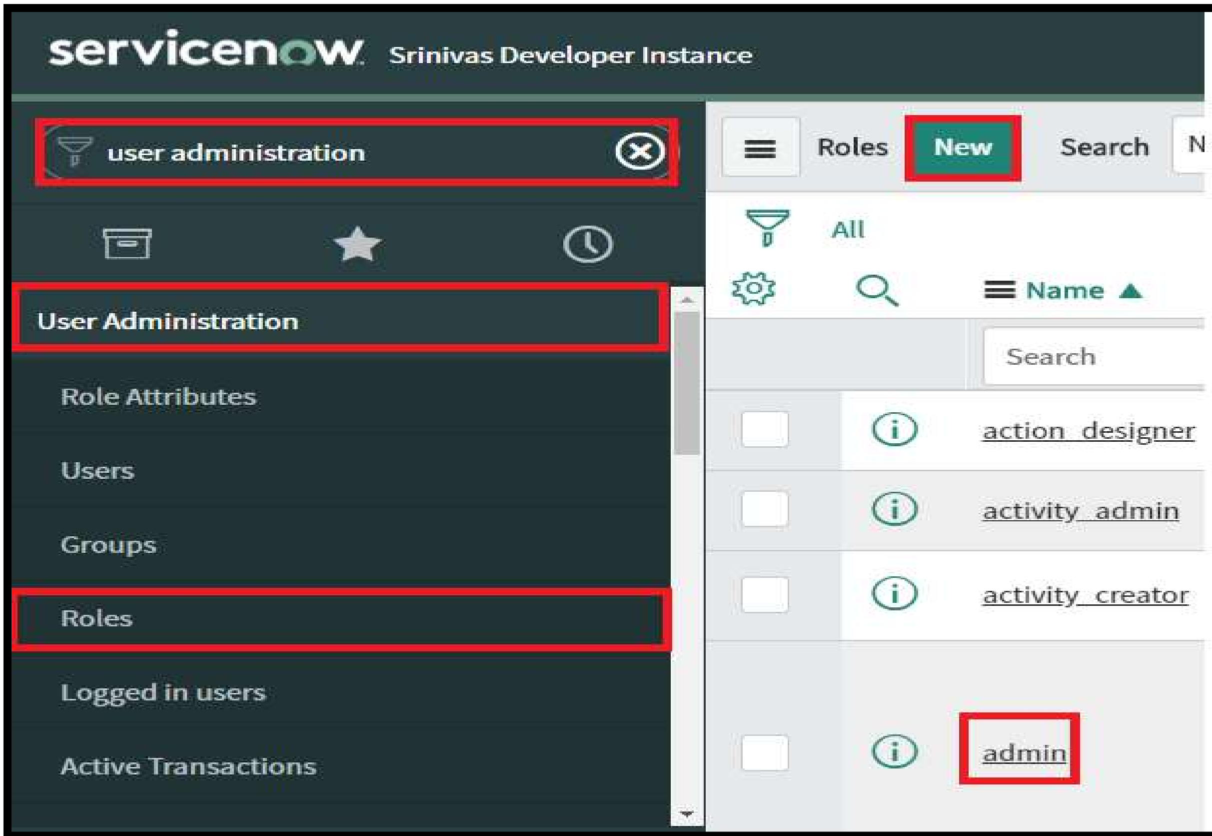
Predefined Roles	
✓ admin	✓ itil
✓ agent_admin	✓ Itil_admin
✓ approval_admin	✓ incident_manager
✓ approver_user	✓ knowledge
✓ asset	✓ knowledge_admin
✓ catalog	✓ major_incident_manager
✓ catalog_admin	✓ maint
✓ catalog_manager	✓ nobody
✓ impersonator	✓ public
✓ problem_manager	✓ report_publisher

Exercise: 8 How to create a new Role

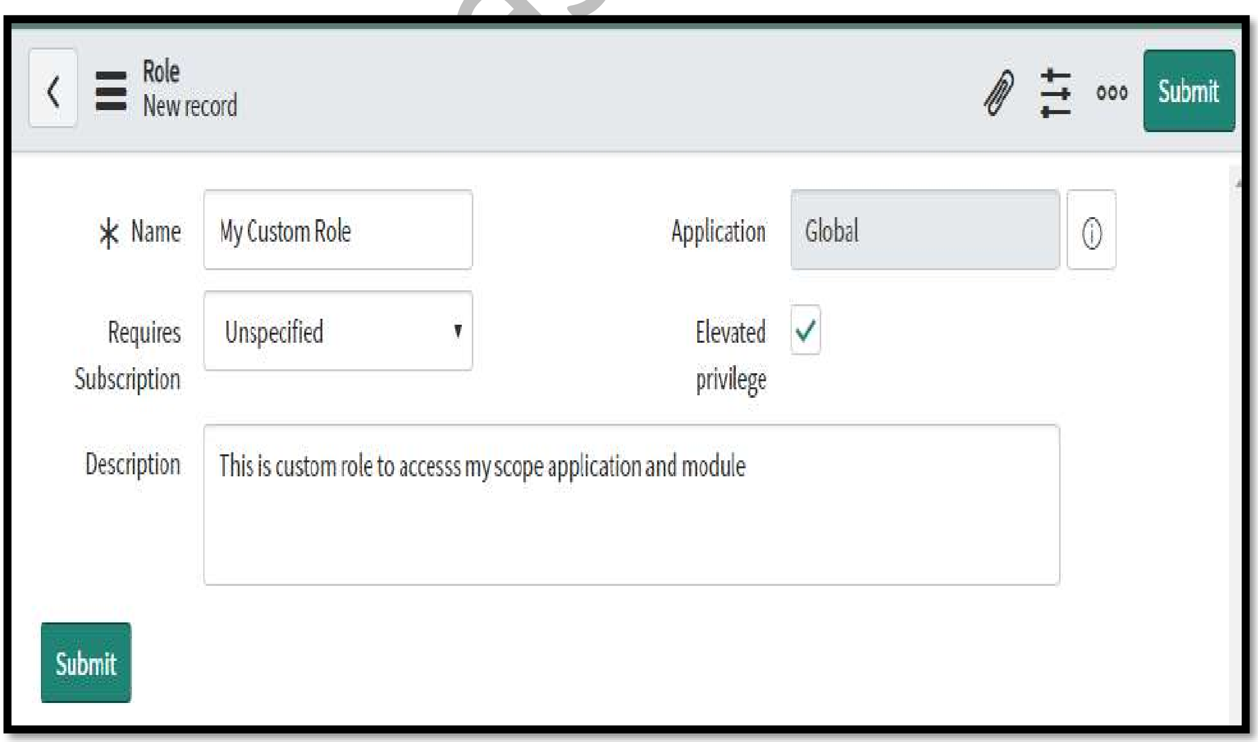
Once access has been granted to a role, all of the groups or users assigned to the role are granted the access. Roles can contain other roles, and any access granted to a role is granted to any role that contains

Procedure

1. Navigate to **User Administration > Roles**
2. Create a **New** record.
3. Fill the form below.



4. Click on **Submit**

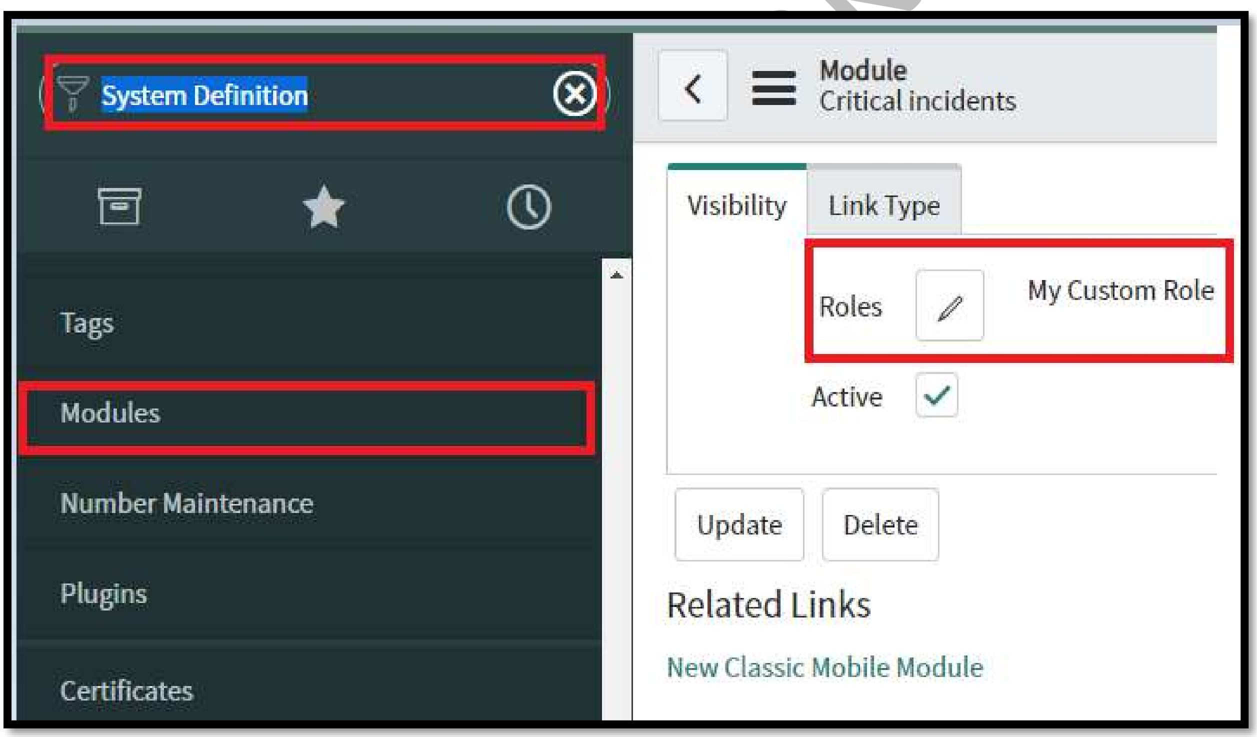


Exercise: 9 Grant a role access to applications and modules

Roles control access to features and capabilities in applications and modules. You add a role to an application or module to enable the role to grant access to the application or module for all users with the role.

Procedure

1. Navigate to **System Definition > Applications** or
2. Navigate to **System Definition > Modules.**
3. Select the particular application or module to open it in form view.
4. Click on lock to open the **Roles field**.
5. Use the slush bucket to add the desired roles to the **application or module.**
6. Click on lock to close the **Roles field**
7. **Save** your changes.



Exercise: 10 Assign a role to User

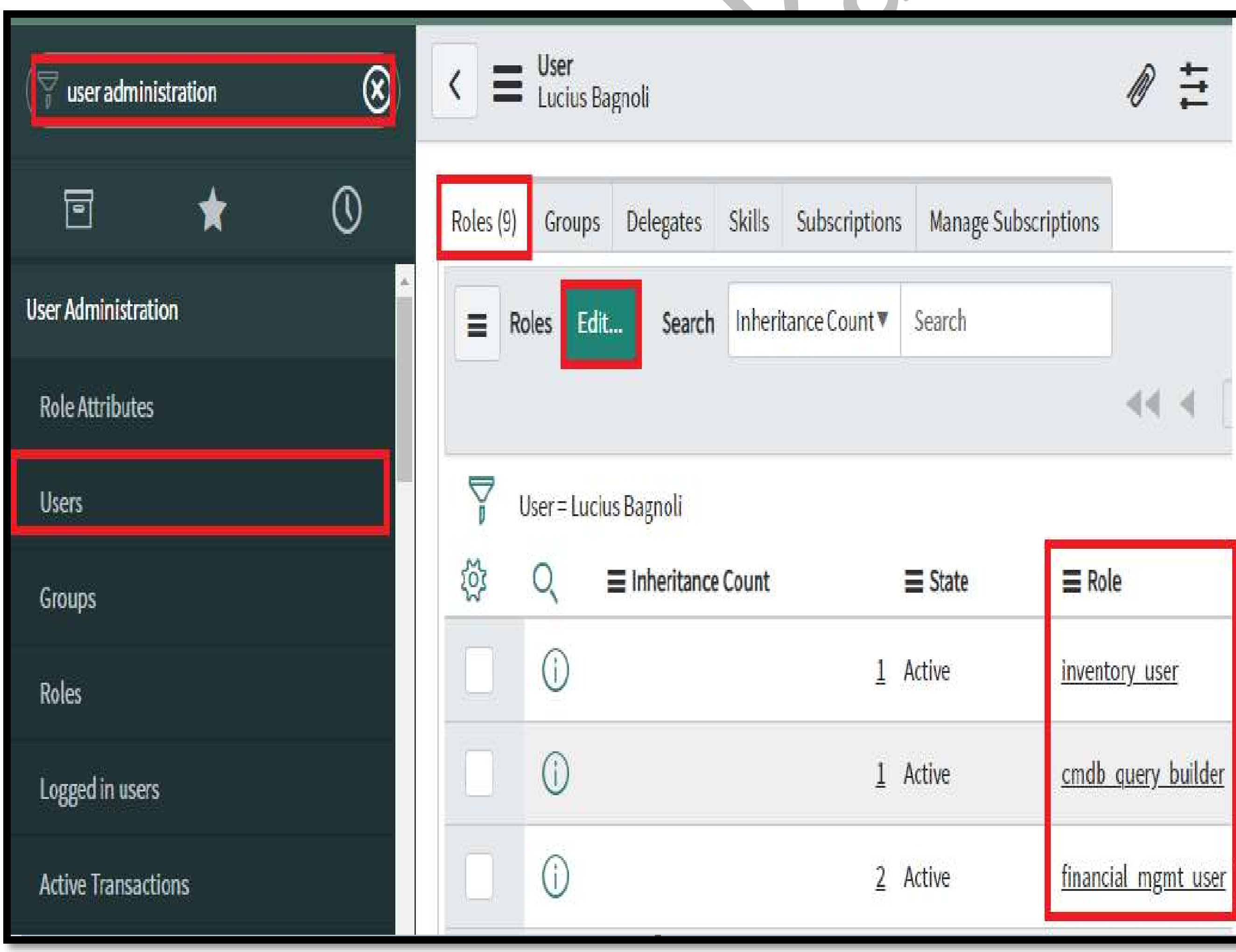
We can assign a role to a user to grant access to **applications and modules**.

A user inherits roles from all groups to which the user belongs. You can also assign roles directly to a user. Whenever a user is assigned a new role, it only takes affect after logging in with a new session.

1. To grant the **admin** role to a user, you must also have the **admin** role.
2. To grant the **security_admin** role to a user, you must also have the **security_admin role**, and must elevate to the **security_admin** role before granting the security_admin role to other users.
   We cannot delete roles that are assigned to the group from the user record. You must remove the user from the group record.

Procedure

1. Navigate to **User Administration > Users**
2. Open any user record.
3. In the **Roles related list**, click **Edit.**
4. In the Collection list, select the desired roles, and then **click Add.**
5. Click on **Update.**



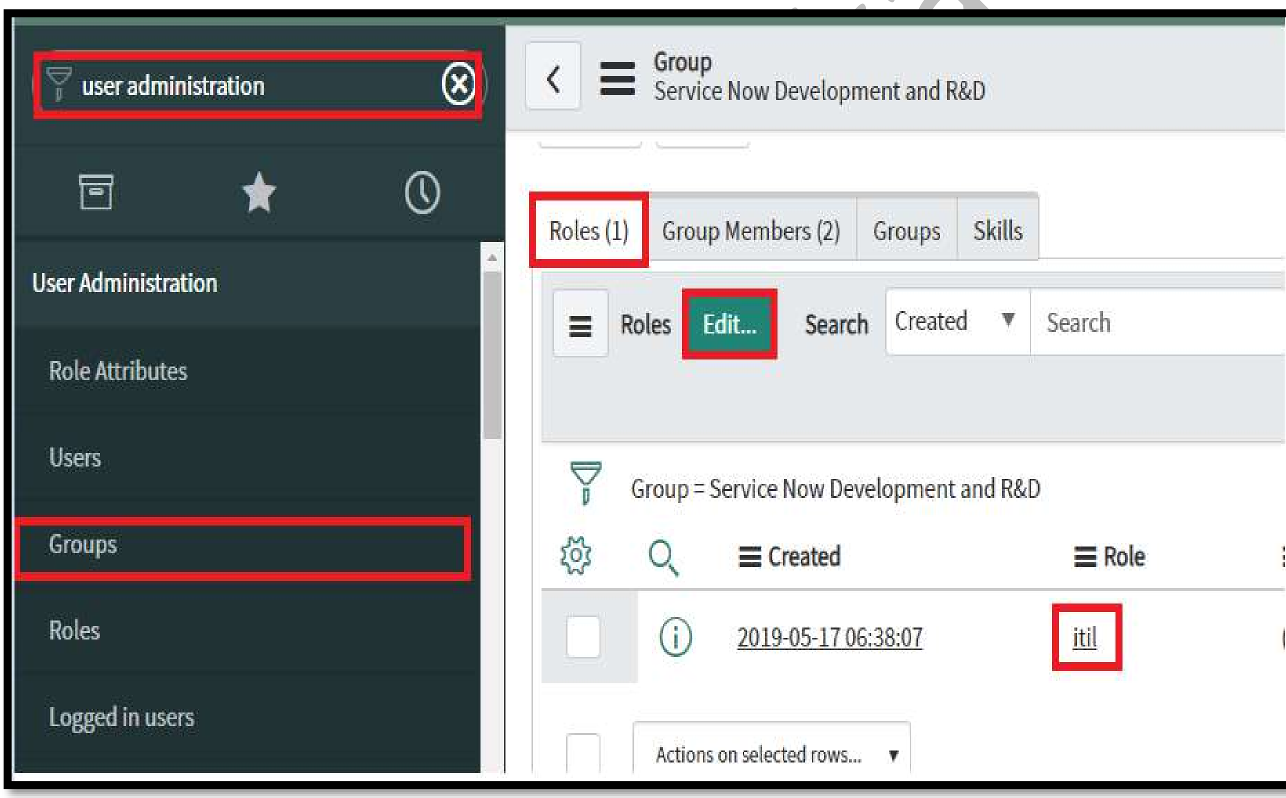
Exercise: 11 Assign a role to a Group

We can assign a role to a **group** to grant access to applications and modules to group of members.

When we assign roles to groups rather than to individual users, members of the group inherit the role. When a user switches groups, the new group role is assigned automatically.

Procedure

1. Navigate to **User Administration > Groups.**
2. Open the any **group** to assign a role.
3. In the **Roles related list**, click **Edit.**
4. Use the slush bucket to add the desired roles to the group.
5. Click **Update**.



Importance of read-only role

The read-only role **(snc_read_only)** restricts a user or a group of users to read-only access on the tables to which the user already has access.

1. This role is not intended to be the only role a user has. It is intended to be an extra role to restrict insert, update, and delete operations on the tables that the user can access as defined by the other roles.

2. After you assign this role to a user, they can no longer can create, update, or delete records on ANY tables.

Note: Assign this role only to users. Do not assign this role to other resources in the system, including applications, ACLs, and so on.

Users with the snc_read_only role have the following restrictions regardless of other roles and privileges they have.

1. User Cannot insert, update, or delete records from the UI or when using the Glide Record API.
2. User Cannot activate or upgrade plugins.
3. User Cannot directly run SQL statements.
4. Cannot upload any XML files.
5. Can only run background scripts when on an instance in the public sandbox environment.

Audit user roles

If we change to user roles are automatically tracked in the Audit Roles **[sys_audit_role]** table.

Interactive users

When we add a new user to the instance then automatically the user become a interactive users. Interactive users can perform the following actions:

1. Use their user name and password to log in to the UI or a service portal.
2. Connect to an instance from a URL that calls a UI page, form, or list, for example, **https://<instance name>. service-now.com/incident.do.**
3. Connect with single sign-on, for example, digest authentication or SAML.
4. Use their credentials to authorize SOAP connections if allowed by strict security.
5. Use their credentials for other API connections such as WSDL, JSON, XML, or XSD without restriction.

Non-interactive users

Non-interactive users can only use their credentials to authorize API connections such as **JSON, SOAP, and WSDL**. They cannot log in to the Service Now UI. The strict security high security setting determines if non-interactive users are subject to **Contextual Security** requirements.

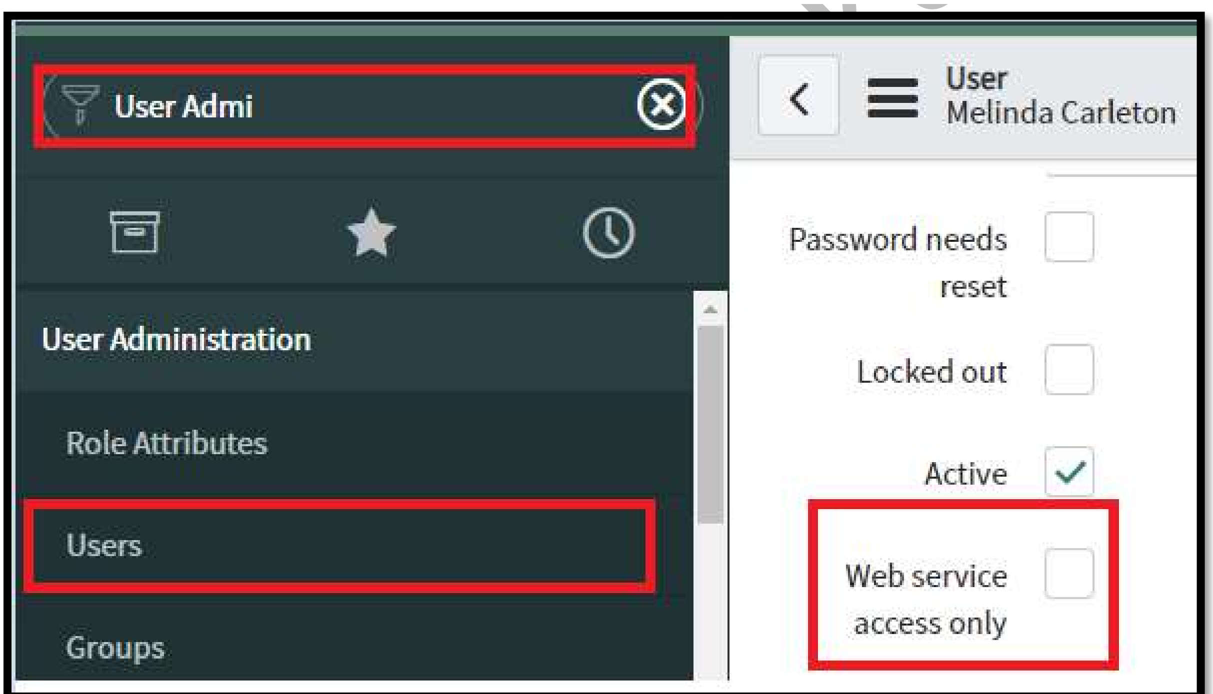
Exercise:12 Create a non-interactive user for web services

Non-interactive users can only connect to a Service Now instance from an API protocol. Use this feature to set up user accounts for **web service authentication purposes**.

Non-interactive users cannot log in to an instance or a service portal, connect through **single-sign-on**, or be used as a **MID Server** user.

Procedure

1. Navigate to **User Administration > Users**.
2. Search for the user to be updated.
3. For example, **SOA**P user.
4. Select the **Web Service Access Only** check box.
5. Click **Update**.



Chapter – 9

UI Policies

What is UI policy and Usage?

UI policies can dynamically change the form behavior of information.

Ui Policy is alternative of **Client Scripts**, its mainly running in client side used to fields make **Read only/Writable, Mandatory/Optional, Visible/Invisible.**

1. We can use UI policies to make the **number field on a form read-only**, make the **short description field mandatory**, and **on hold reason filed is Hide**.
2. Basic UI policies do not require any scripting,
3. however, for more advanced actions, in case if you want use scripting enable option **Run scripts**
4. You can also use client scripts to perform all of these actions, but for faster load times use UI policies when possible.
5. UI policies change fields on a form based on a set of conditions.
6. Use UI policies to show or hide fields, or to make fields mandatory or read-only based on these conditions.
7. A UI policy specifies one or more actions to take when the policy is executed.
8. First we can create the ui policy. Then add as many actions as required.

UI policy actions are applied only if all of the following conditions are met:

- ✓ The UI Policy record is **Active**
- ✓ The items in the **Conditions** field evaluate to true
- ✓ The field specified in the **UI policy action** is present on the specified form

Creating a UI policy

We can create a **UI policy** to implement the conditions in the Incident form when the stage was changes to different states

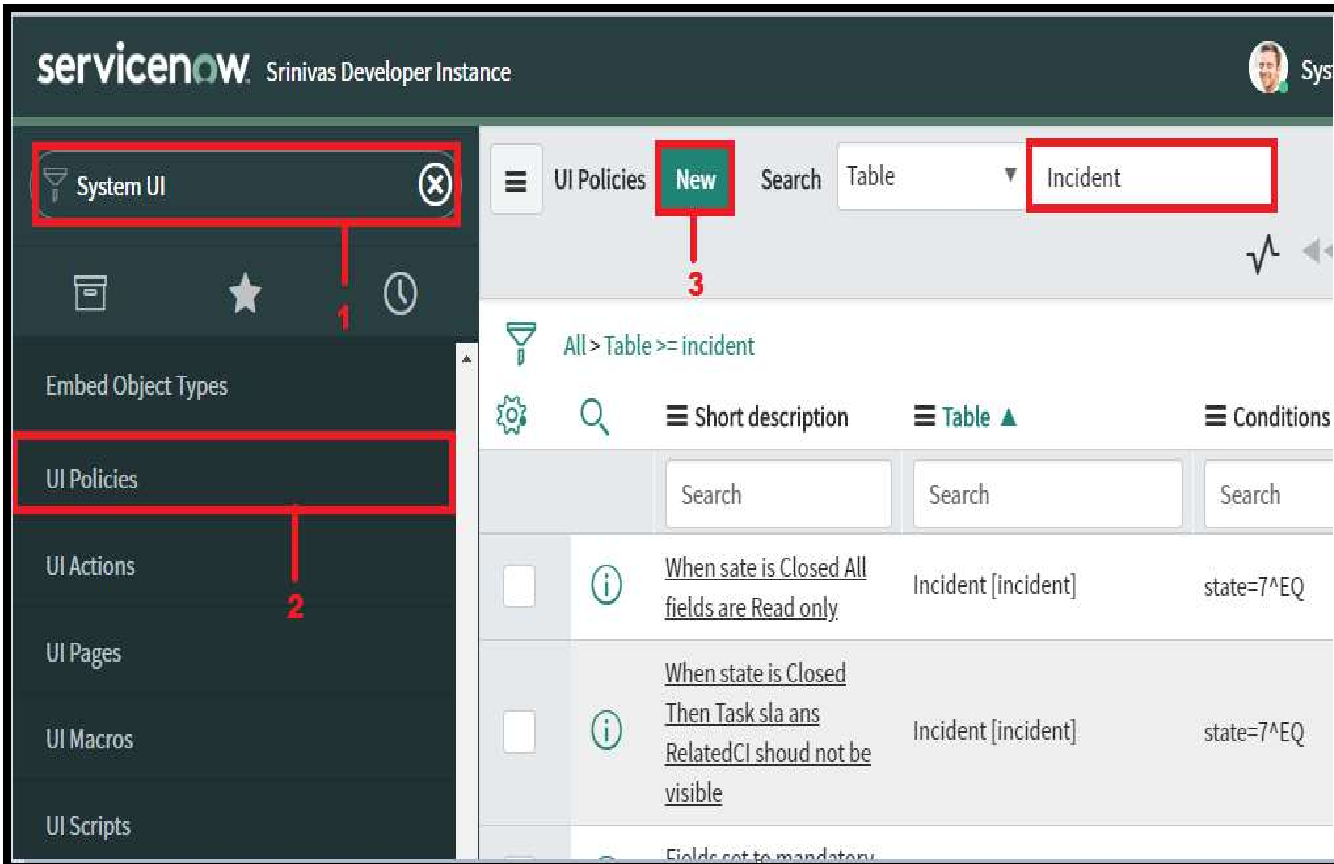
Exercise: 1 About This Requirement

Number and **Priority** field are read only and **Caller** and **Short Description** fields are mandatory for all states in **[Incident] Table**

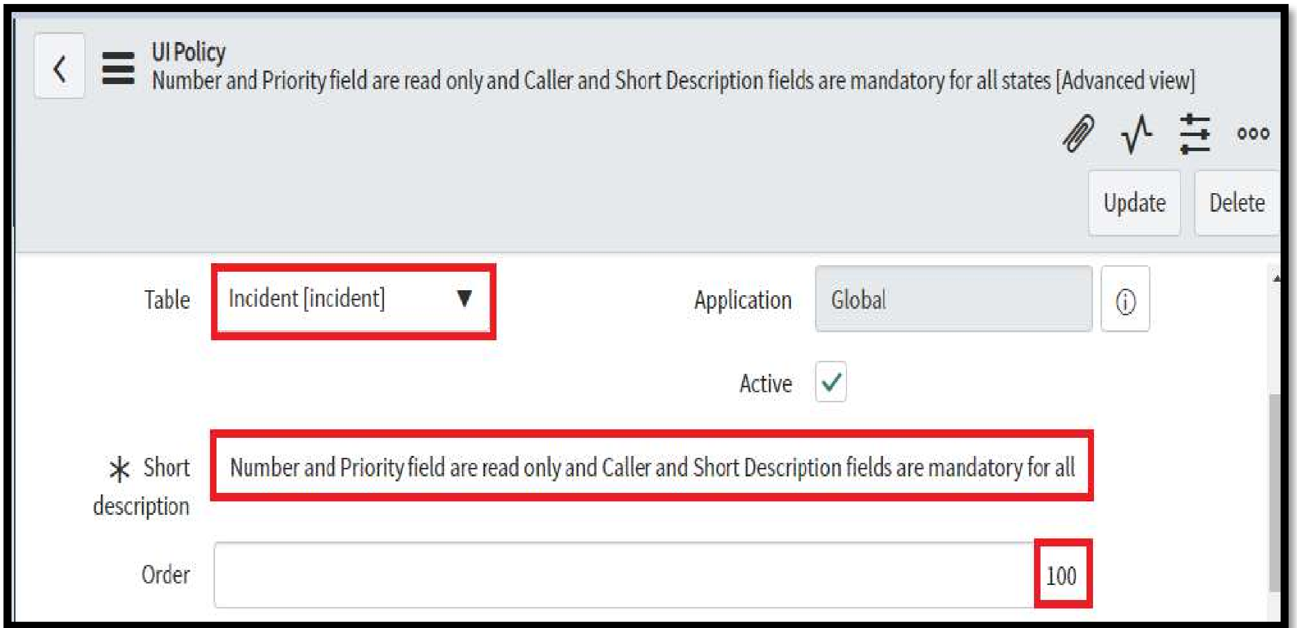
- **Number** and **Priority** Fields are **Read Only**
- **Caller** and **Short Description** fields are **Mandatory**

Procedure

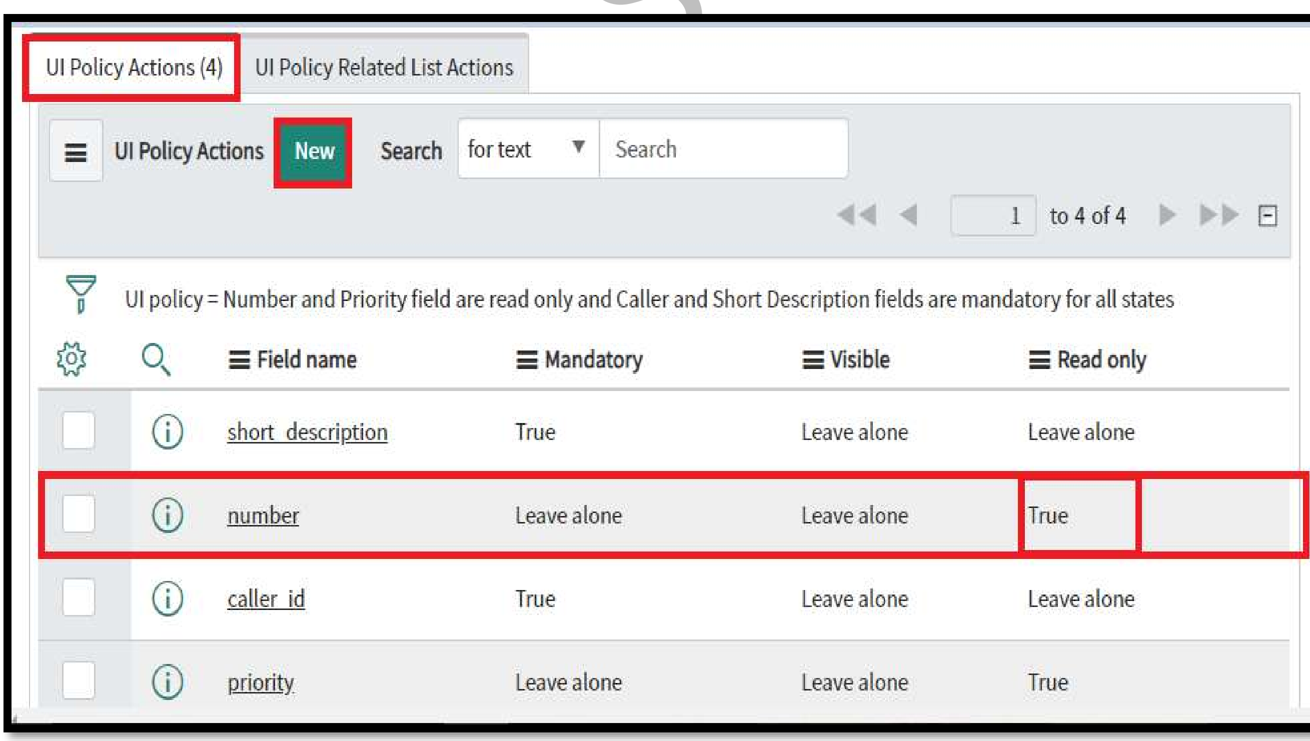
1. Navigate to **System UI** > **UI Policies**.
2. Click **New**.
3. Fill the form with below details



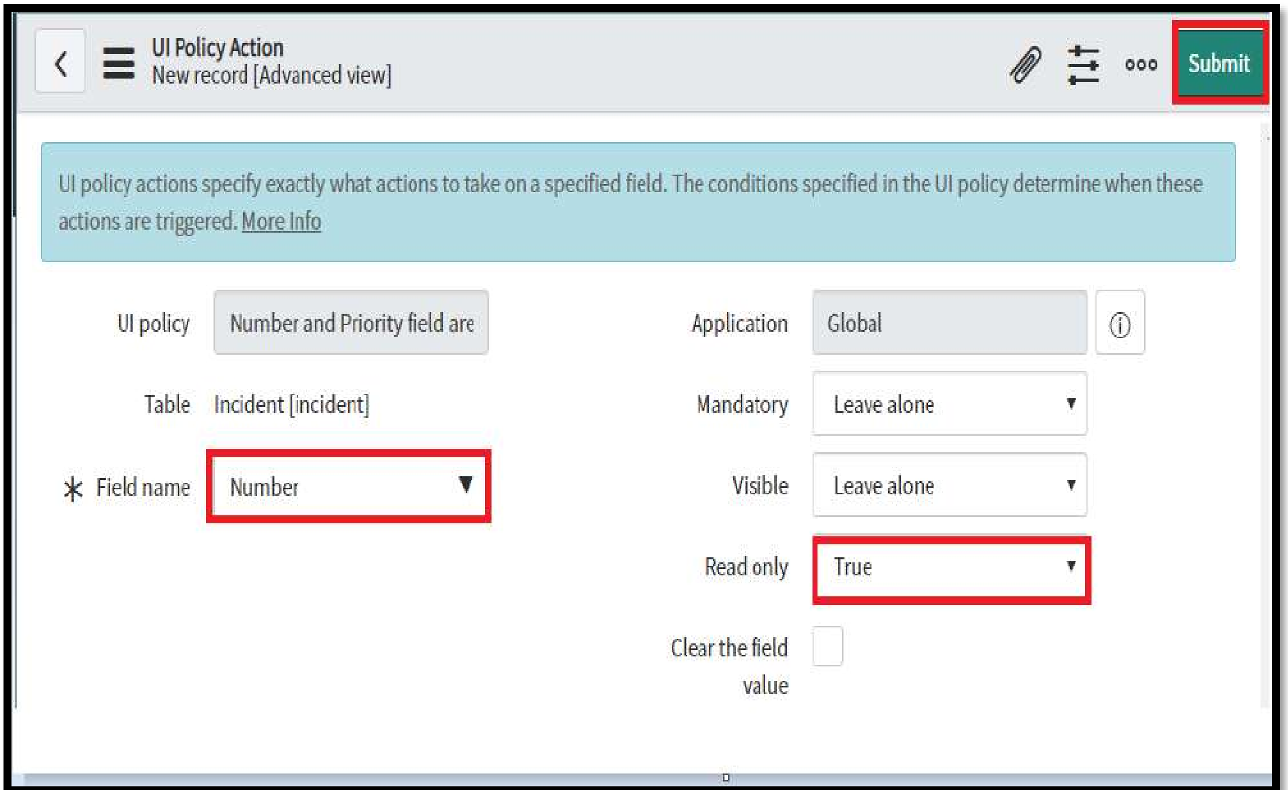
4. **Table:** Incident
5. **Order** :**100**
6. **Short Description: Number and Priority** field are read only and **Caller and Short Description** fields are mandatory for all states in **[Incident] Table**
7. Click on **Submit**



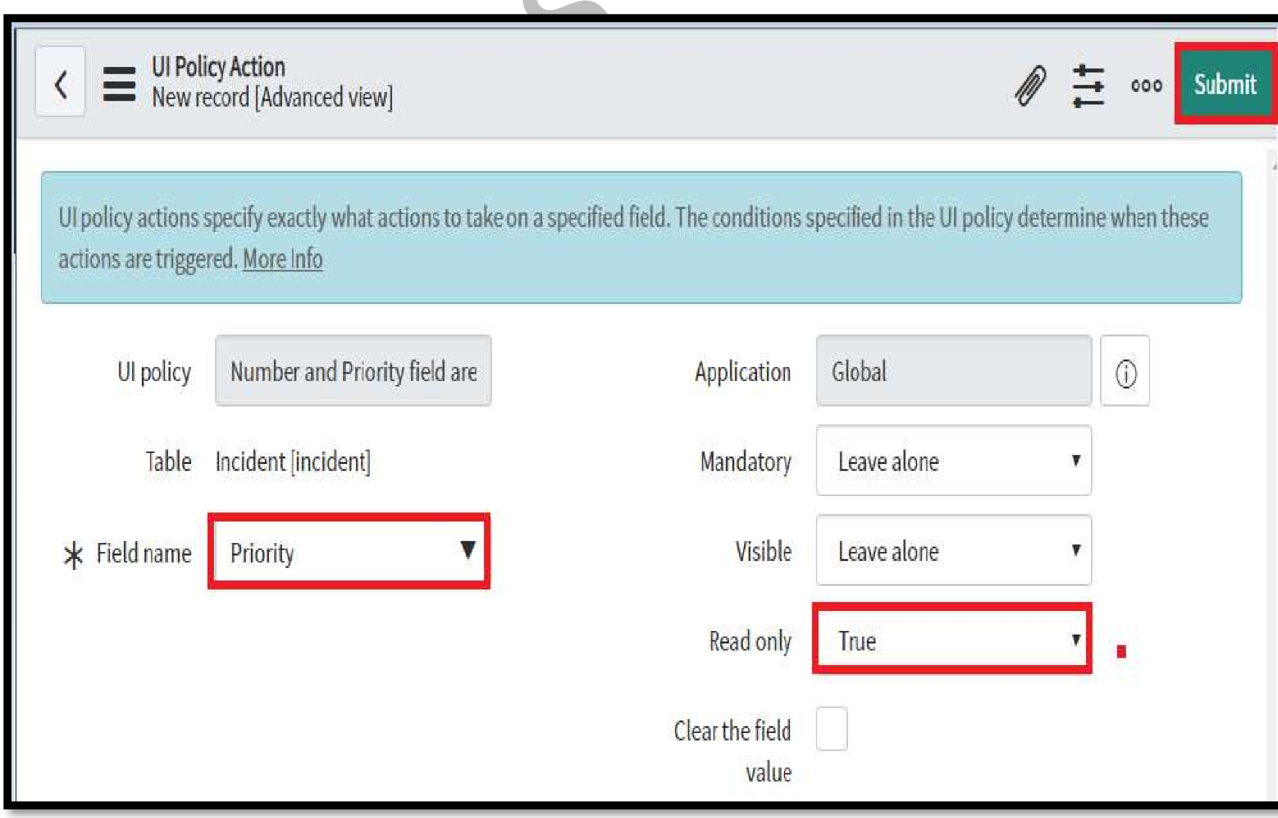
8. Again open the same **UI policy record** and apply action
9. Go to **UI Policy Action** tab from **Related List**
10. Click on UI Policy Action **New**



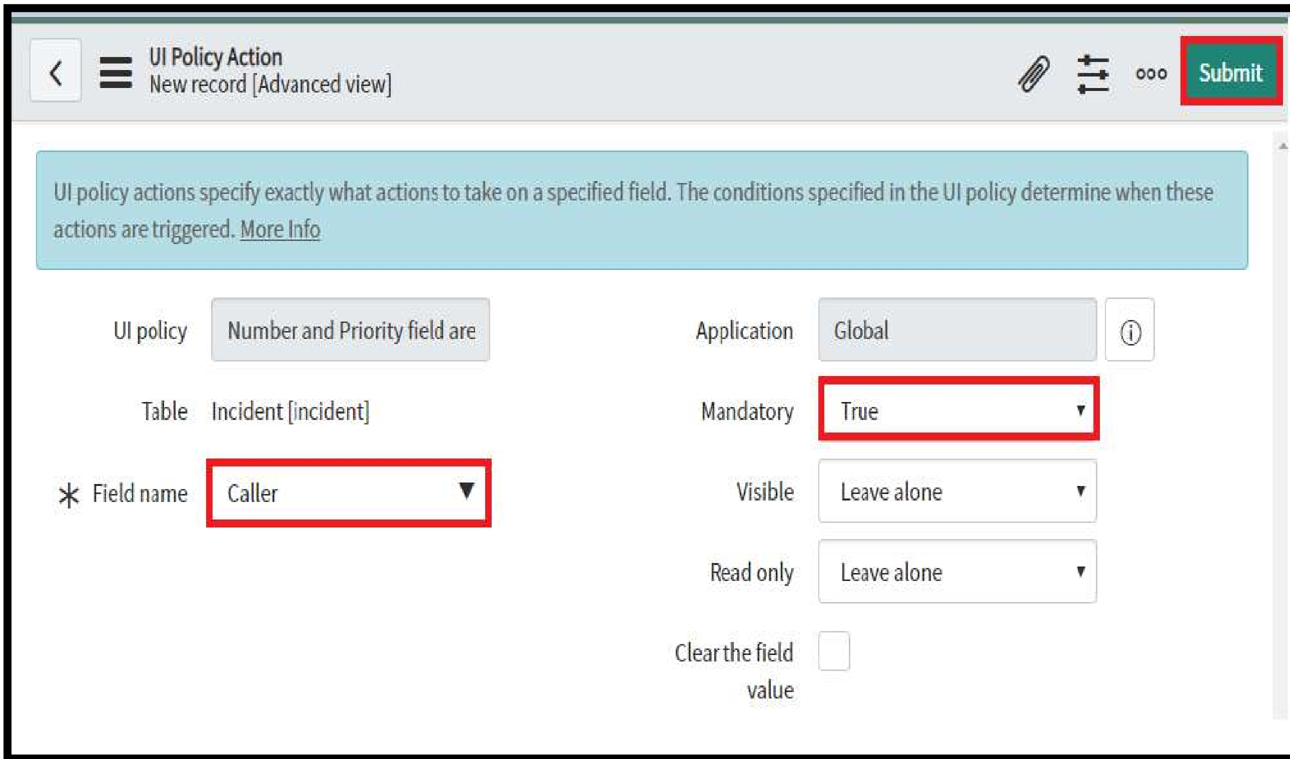
11. Select **Number** field make it **read only true**
12. Click on **Submit**



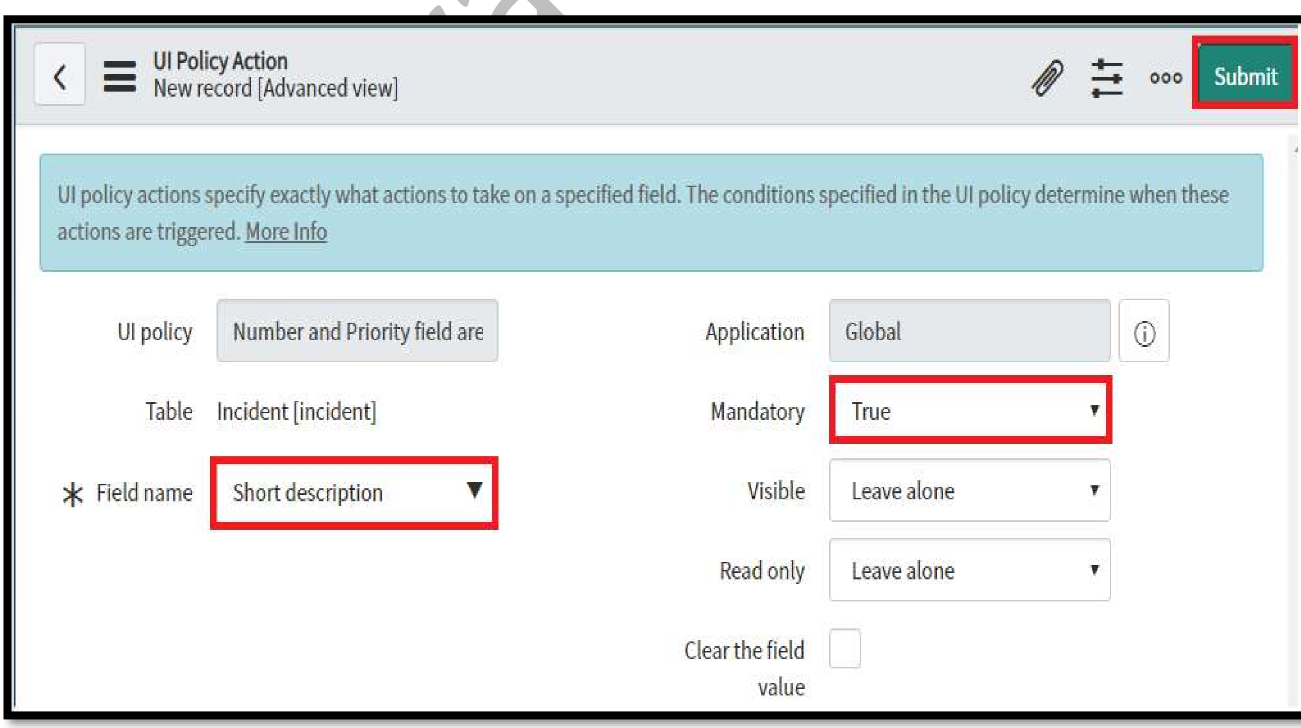
13. Similarly set to remaining fields **Priority** Read Only **True**



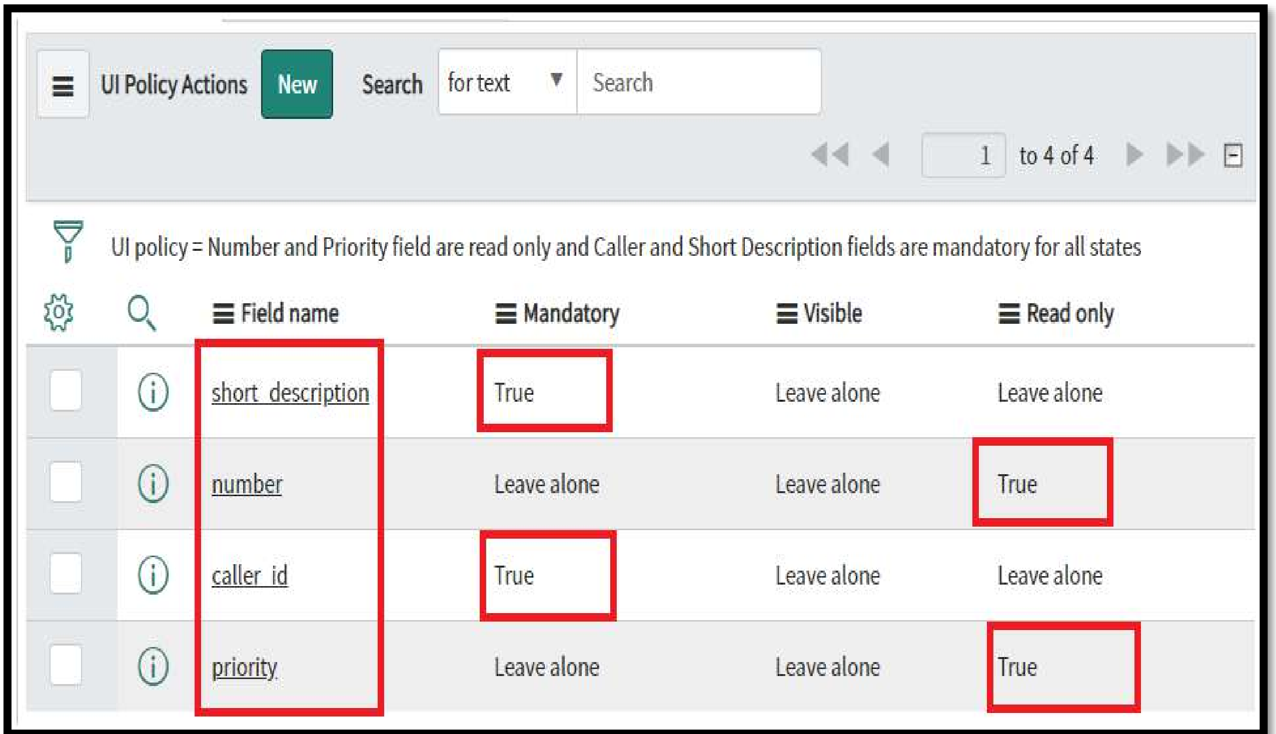
14. Caller Mandatory **True**



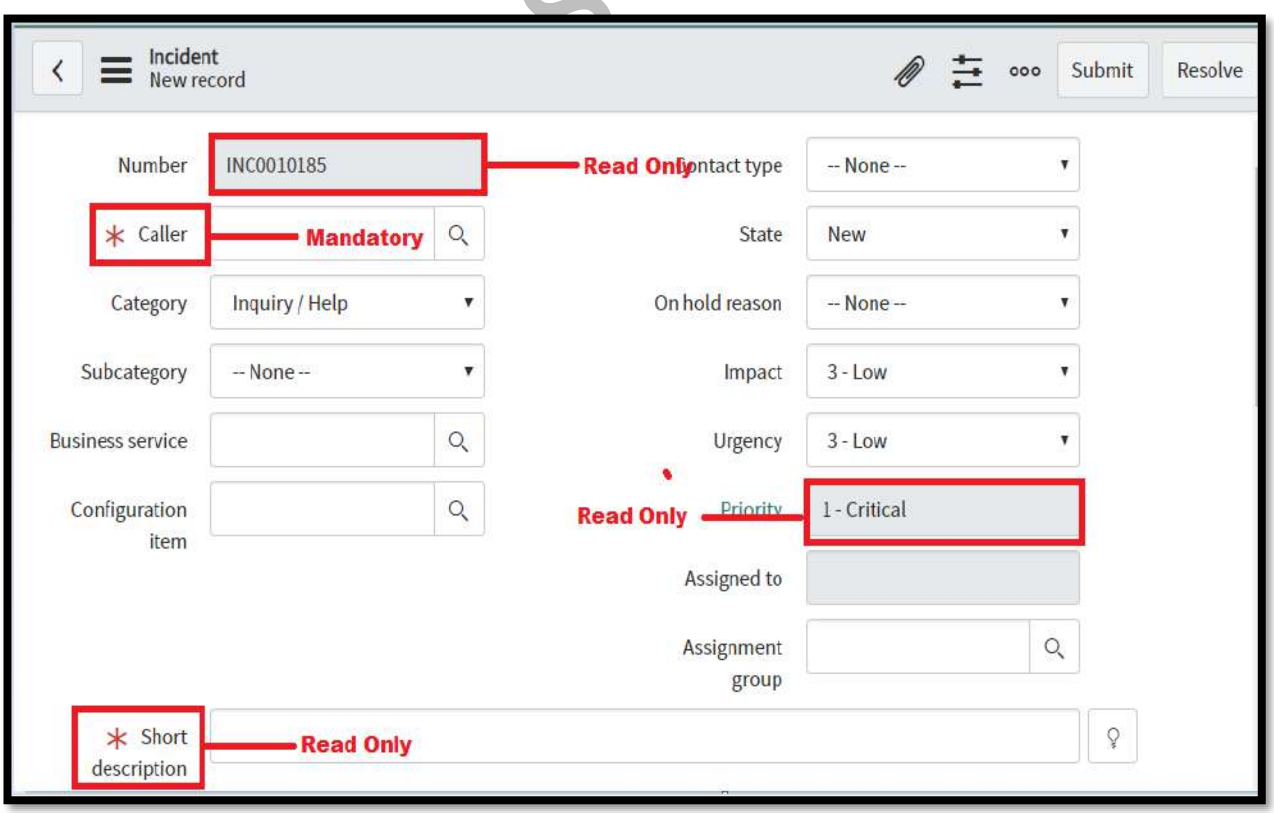
15. **Short Description** Mandatory **True**



16. Follow the below screen shots to find all actions in UI Policy



17. Check in Incident records how these are working



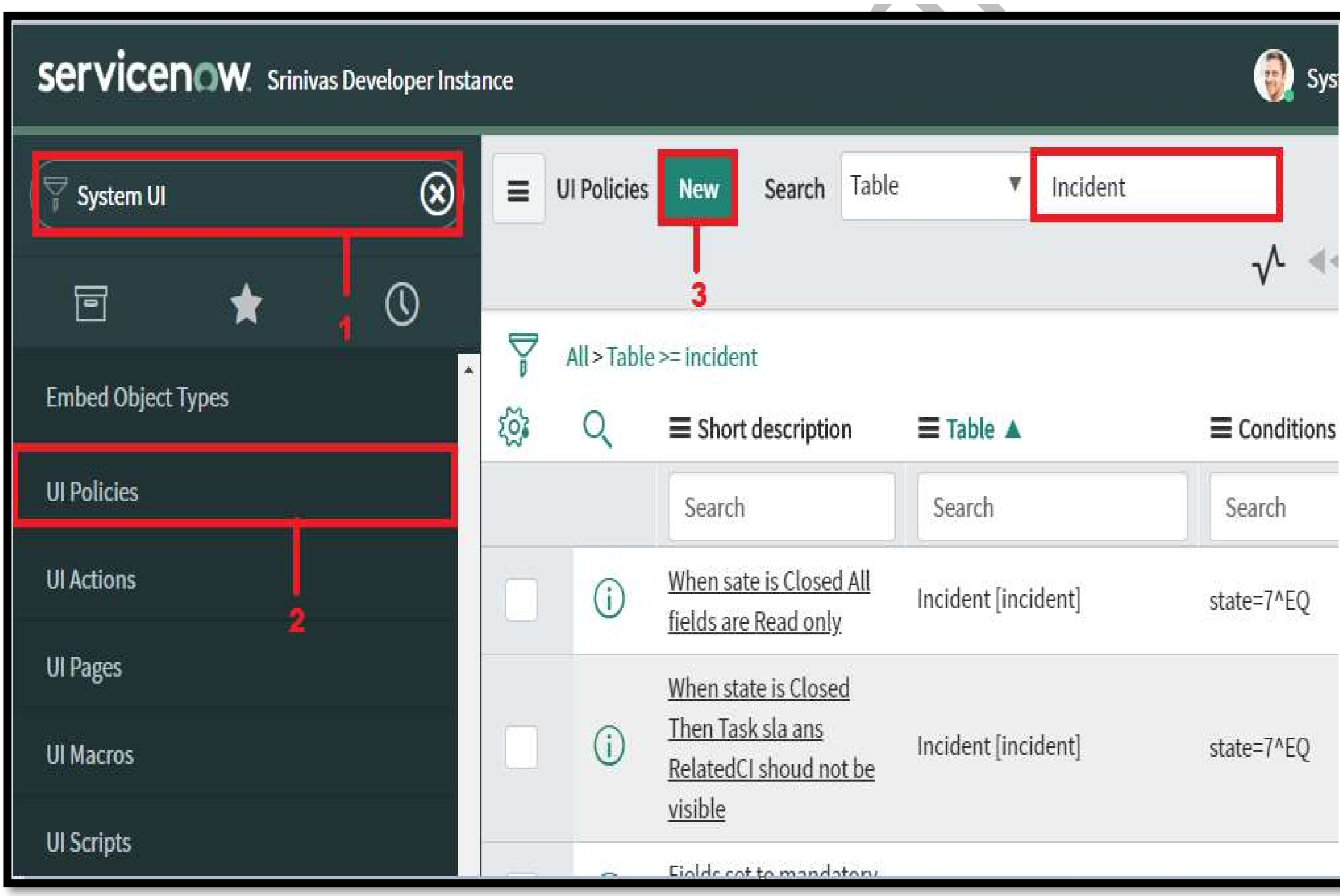
Exercise: 2 About This Requirement

When the Incident State is **On-Hold** Then **On Hold Reason** field should be **Visible** and **Mandatory**

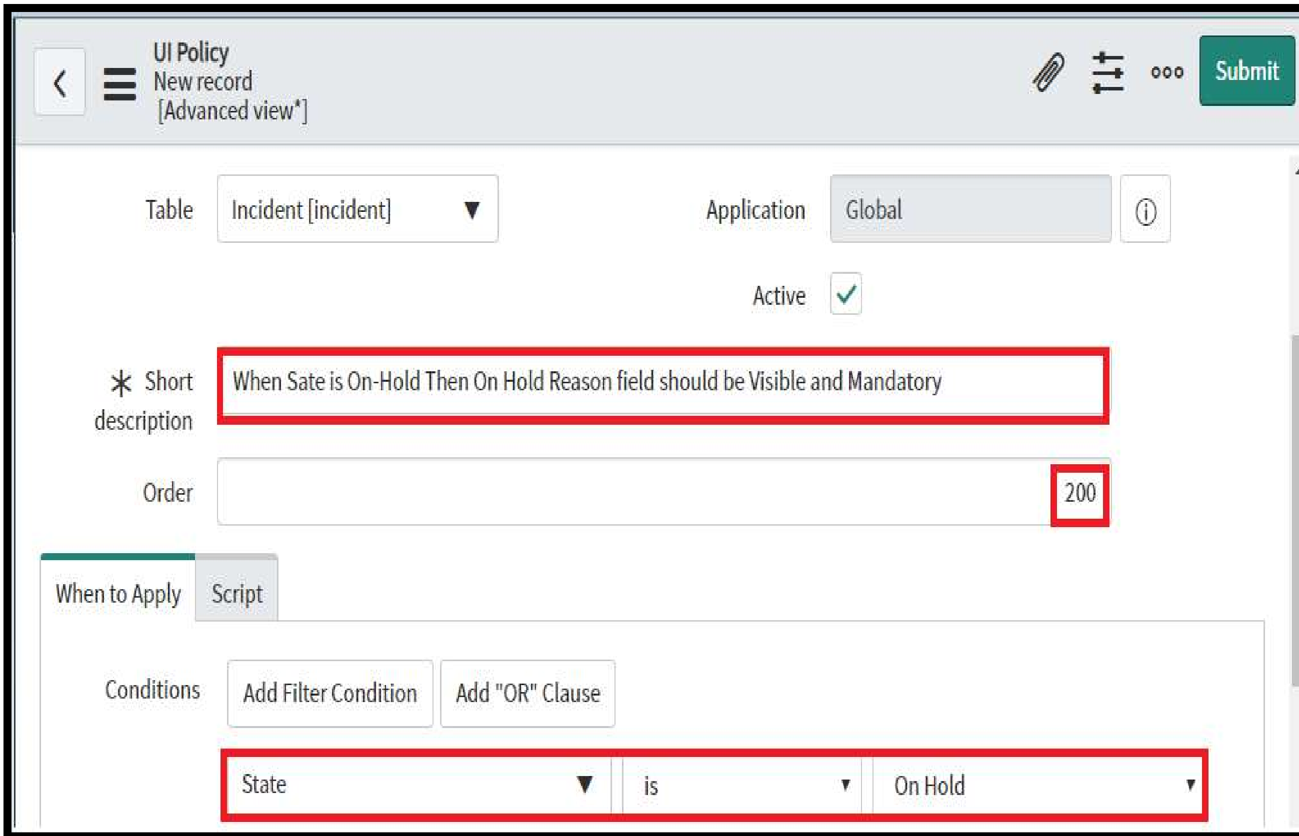
- **Condition** : **[State] [Is] [On Hold]**
- **Action** : **On-Hold Reason** Filed is **Visible True** / **Mandatory True**

Procedure

1. Navigate to **System UI** > **UI Policies**.
2. Click **New**.
3. Fill the form with below details



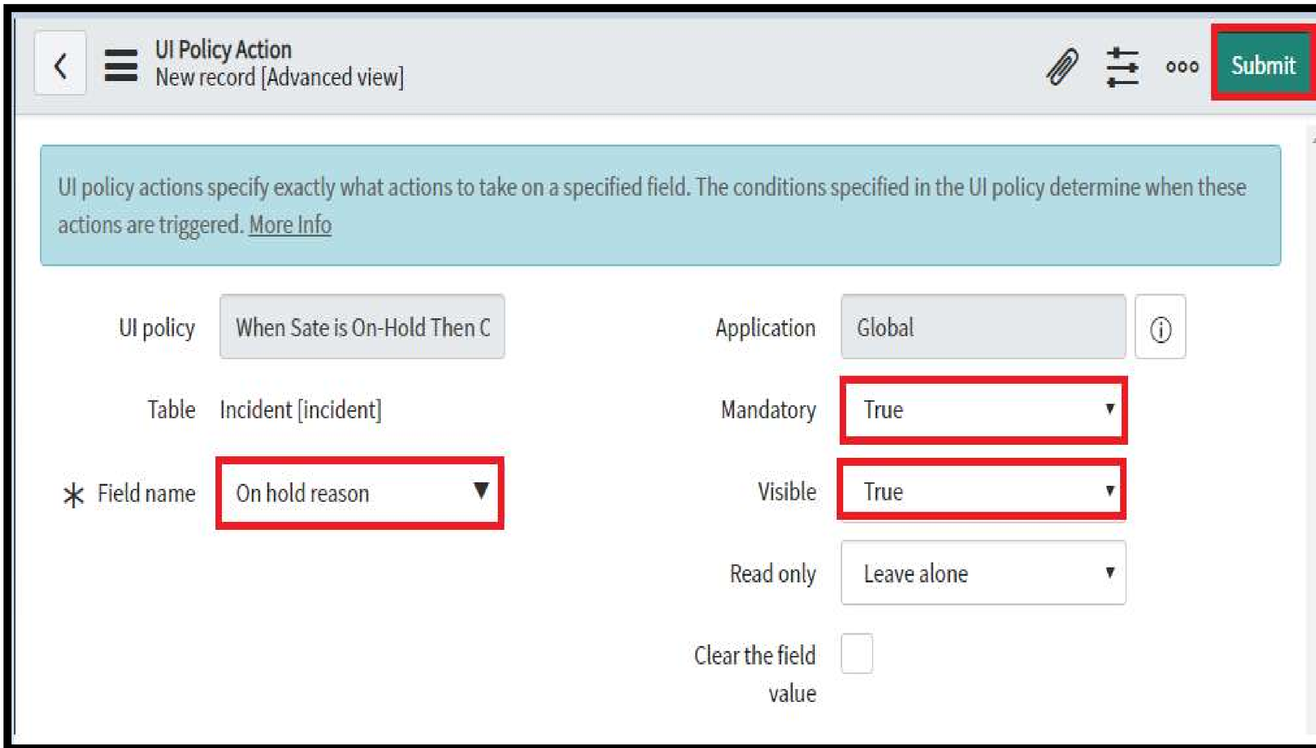
4. **Table:** Incident
5. **Conditions:** state is **On-Hold**
6. **Order** :**200**
7. **Short Description:** When Incident Sate is **On-Hold** Then **On Hold Reason** field should be **Visible and Mandatory**
8. Click on **Submit**



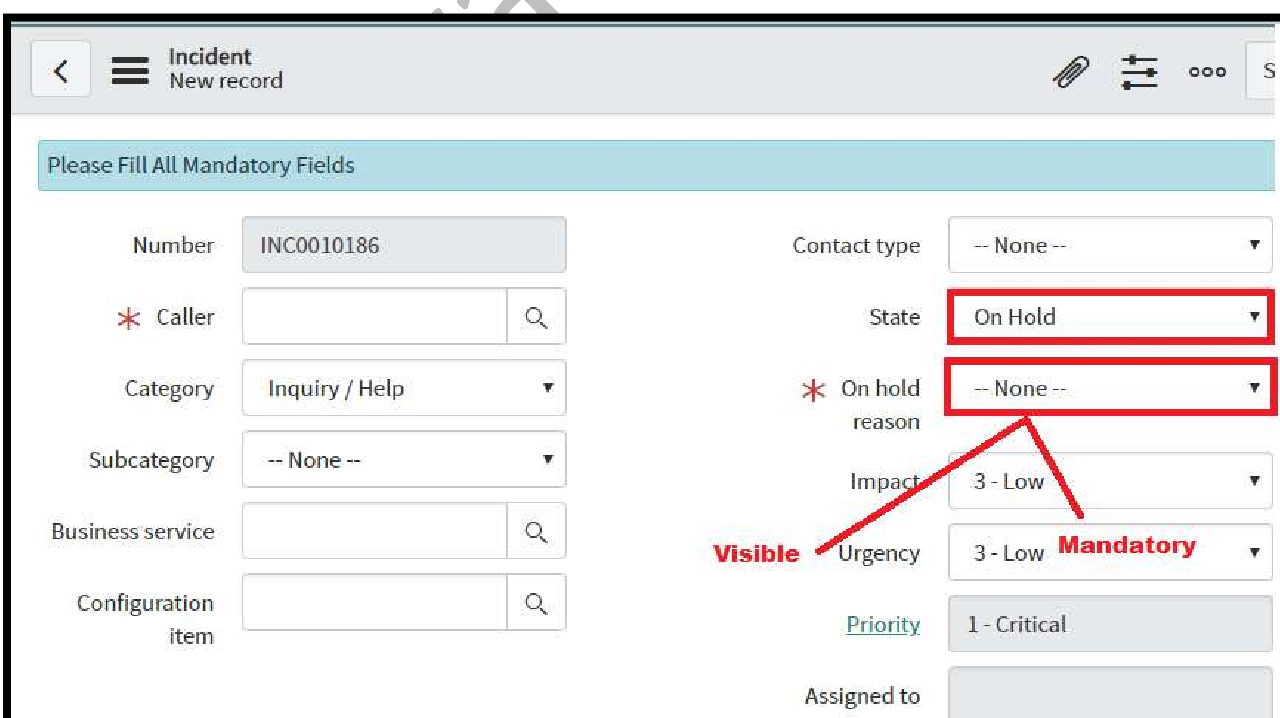
9. Again open the same **UI policy record** and apply action
10. Go to **UI Policy Action** tab from **Related List**
11. Click on UI Policy Action **New**

12. Set On hold reason filed is **Mandatory and Visible True**

13. Click on **Submit**



14. Check in **Incident records** how it is working



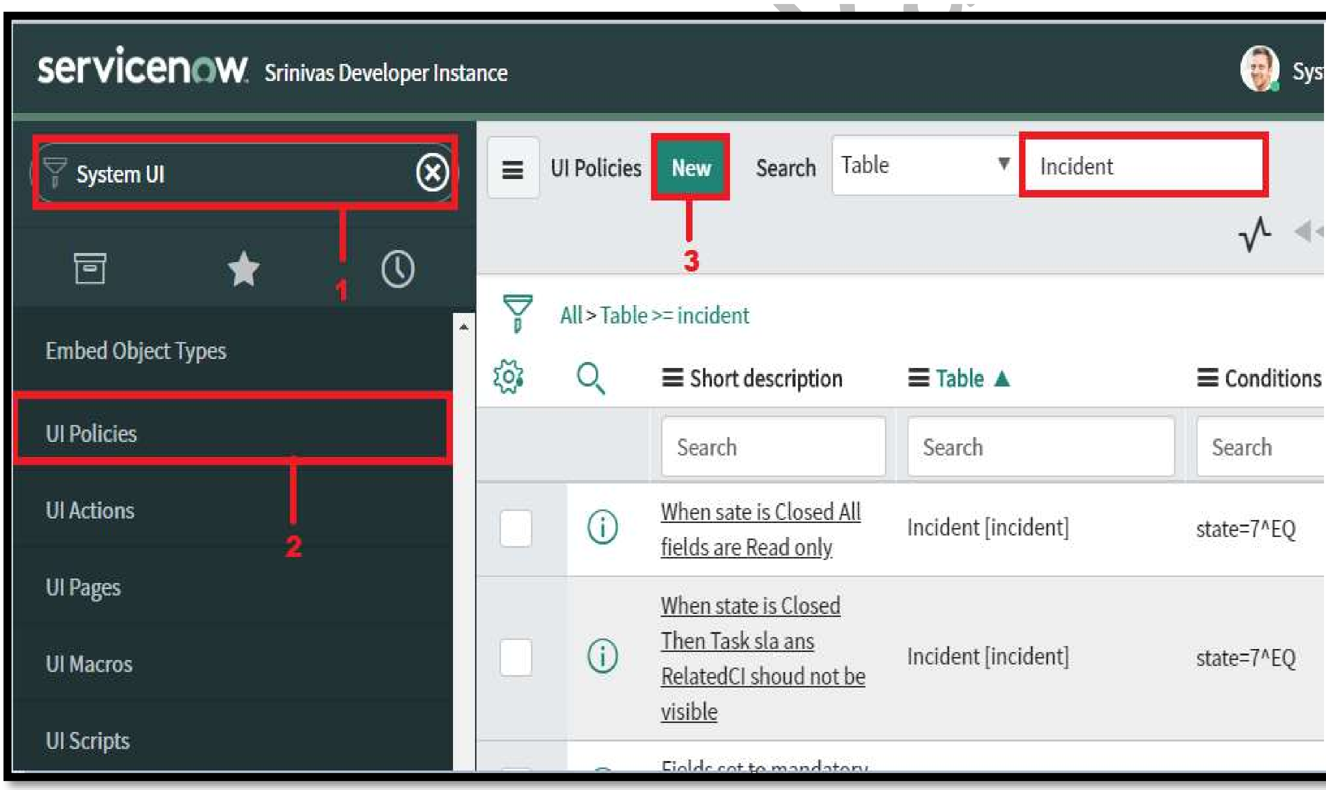
Exercise: 3 About This Requirement

When the Incident state is **Resolved** Then **Assigned to** and **Assignment group** fields are should not be visible

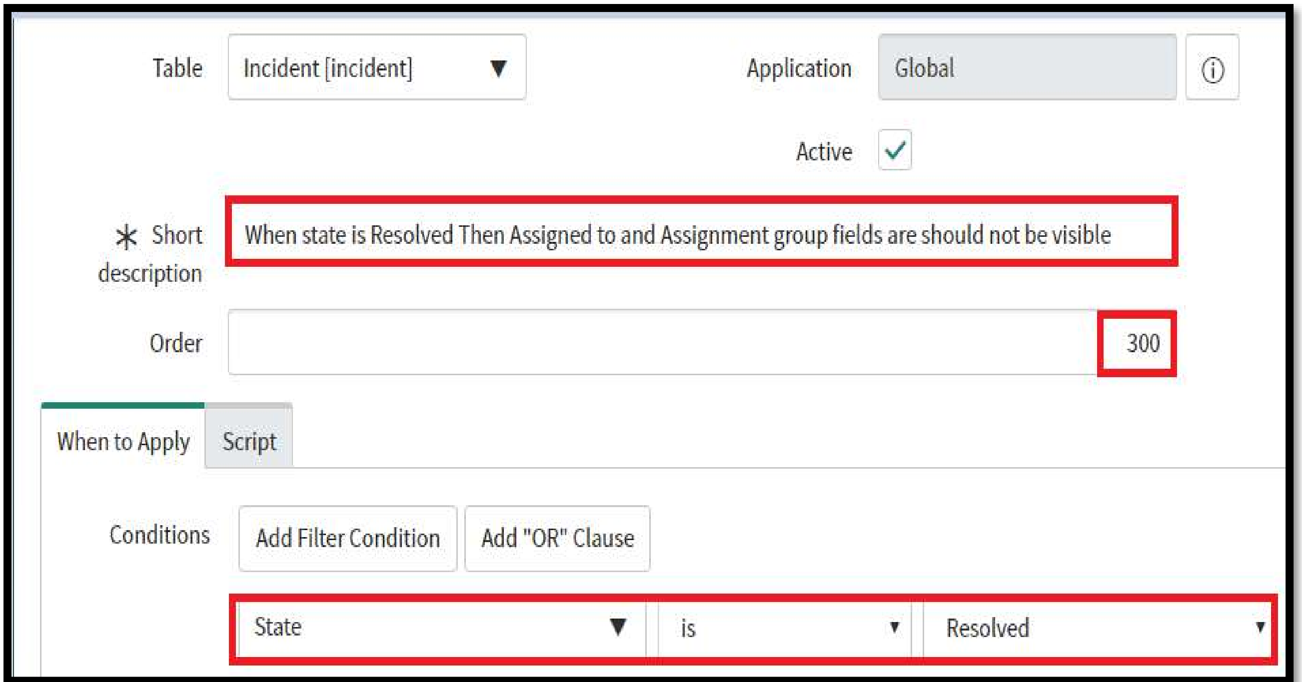
- **Condition** **: [State] [Is] [Resolved]**
- **Action** **: Assigned to** Visible **False**
- **Action** **: Assignment group** Visible **False**

Procedure

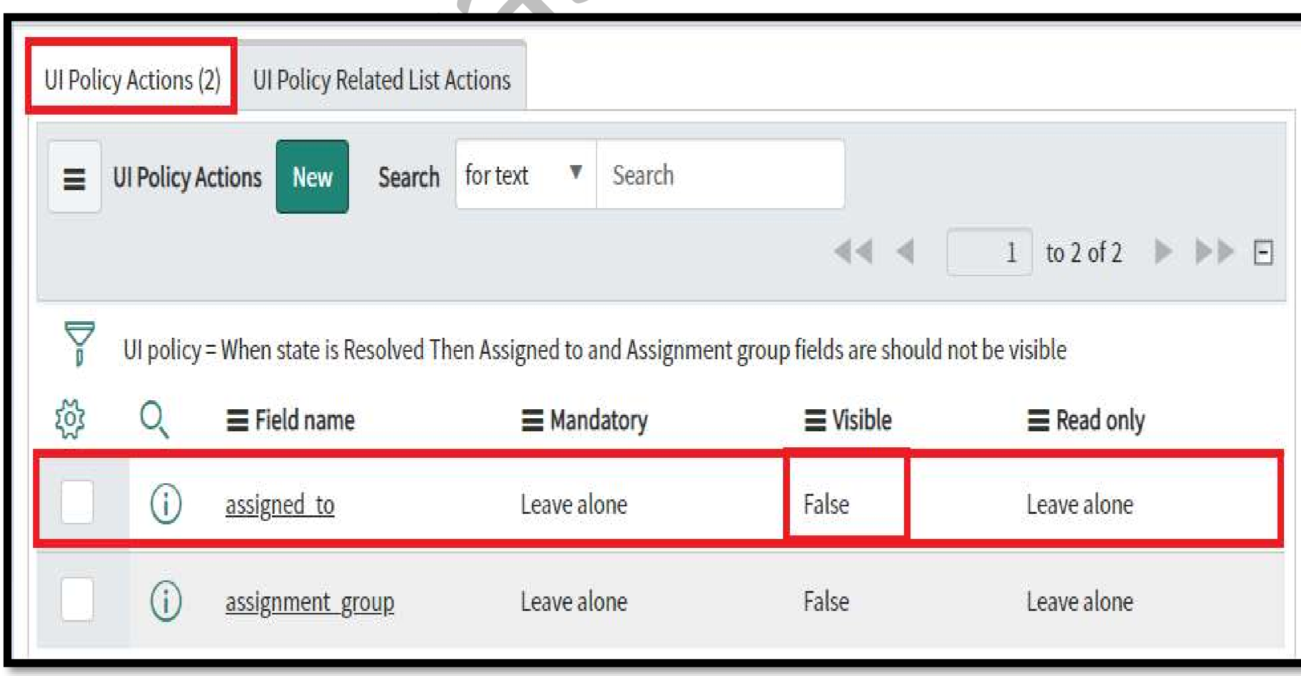
1. Navigate to **System UI** > **UI Policies**.
2. Click **New**.
3. Fill the form with below details



4. **Table:** Incident
5. **Conditions:** state is **Resolved**
6. **Order** :**300**
7. **Short Description:** When The Incident state is **Resolved** Then **Assigned to** and **Assignment group** fields are should not be visible
8. Click on **Submit**

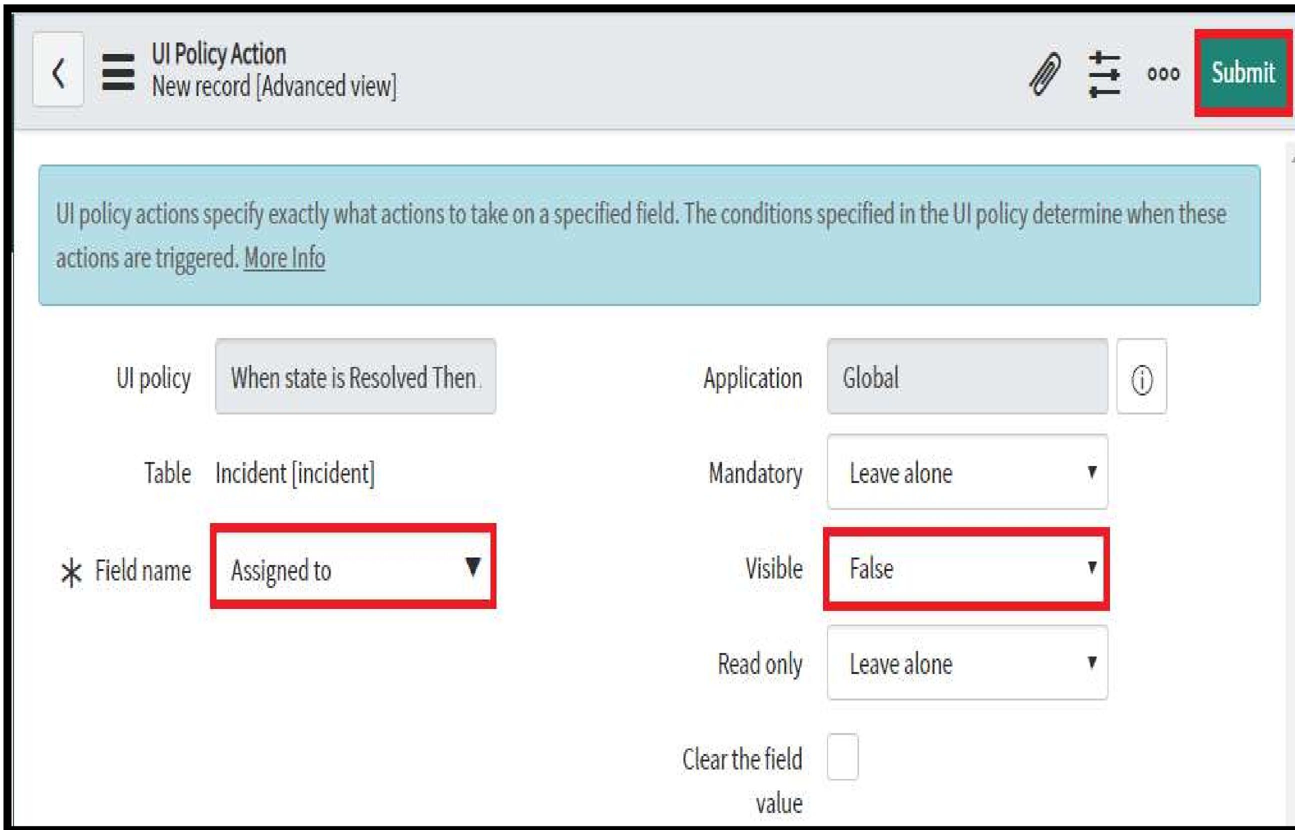


9. Again open the same **UI policy record** and apply action
10. Go to **UI Policy Action** tab from **Related List**
11. Click on UI Policy Action **New**
12. Assign to visible **False**
13. Click on **Submit**

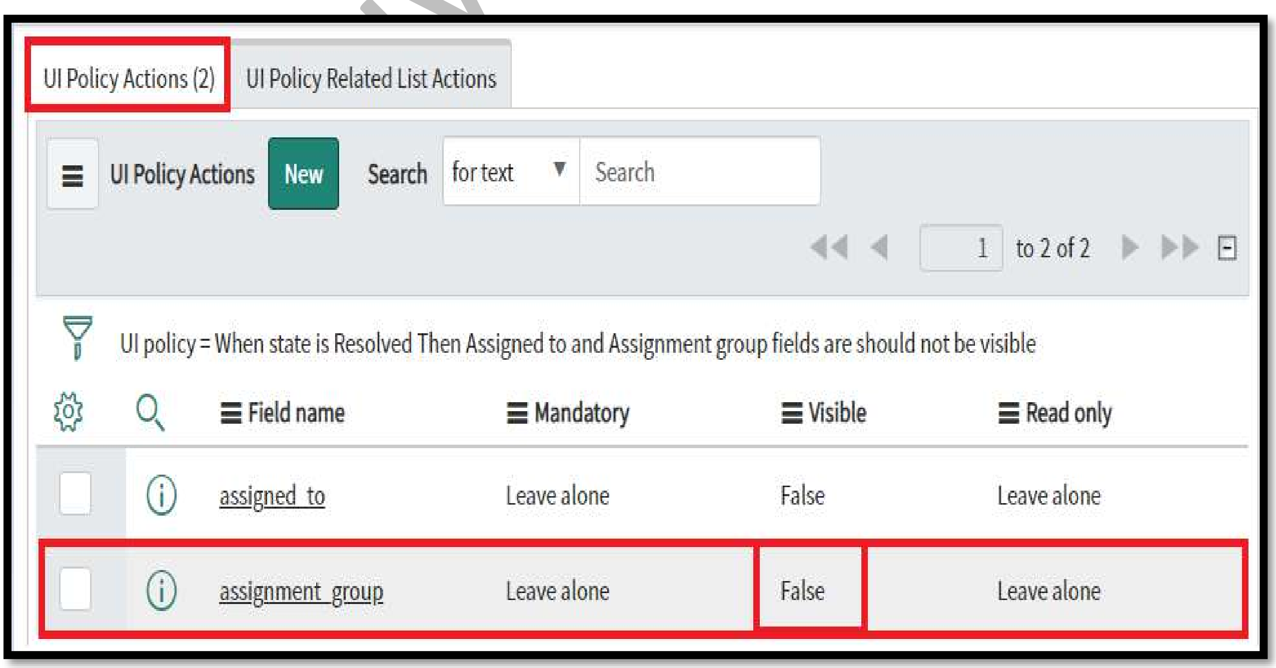


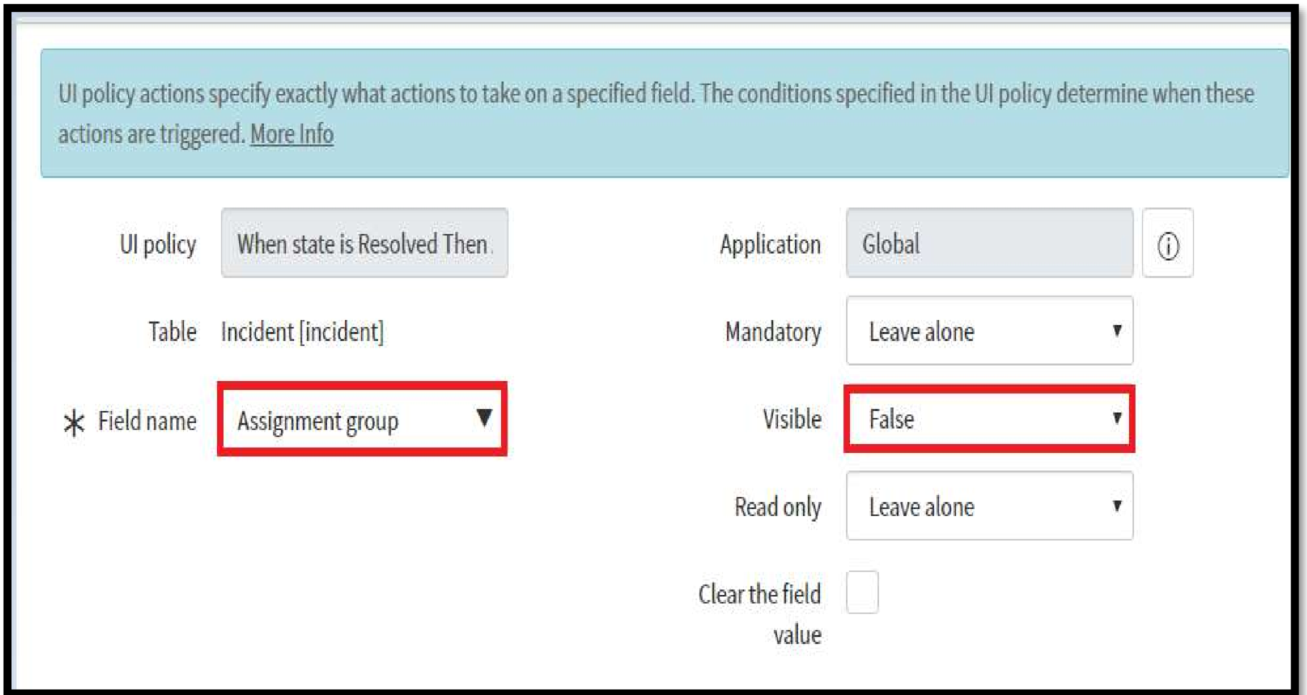
14. Assigned to Visible **False**

15. Click on **Submit**

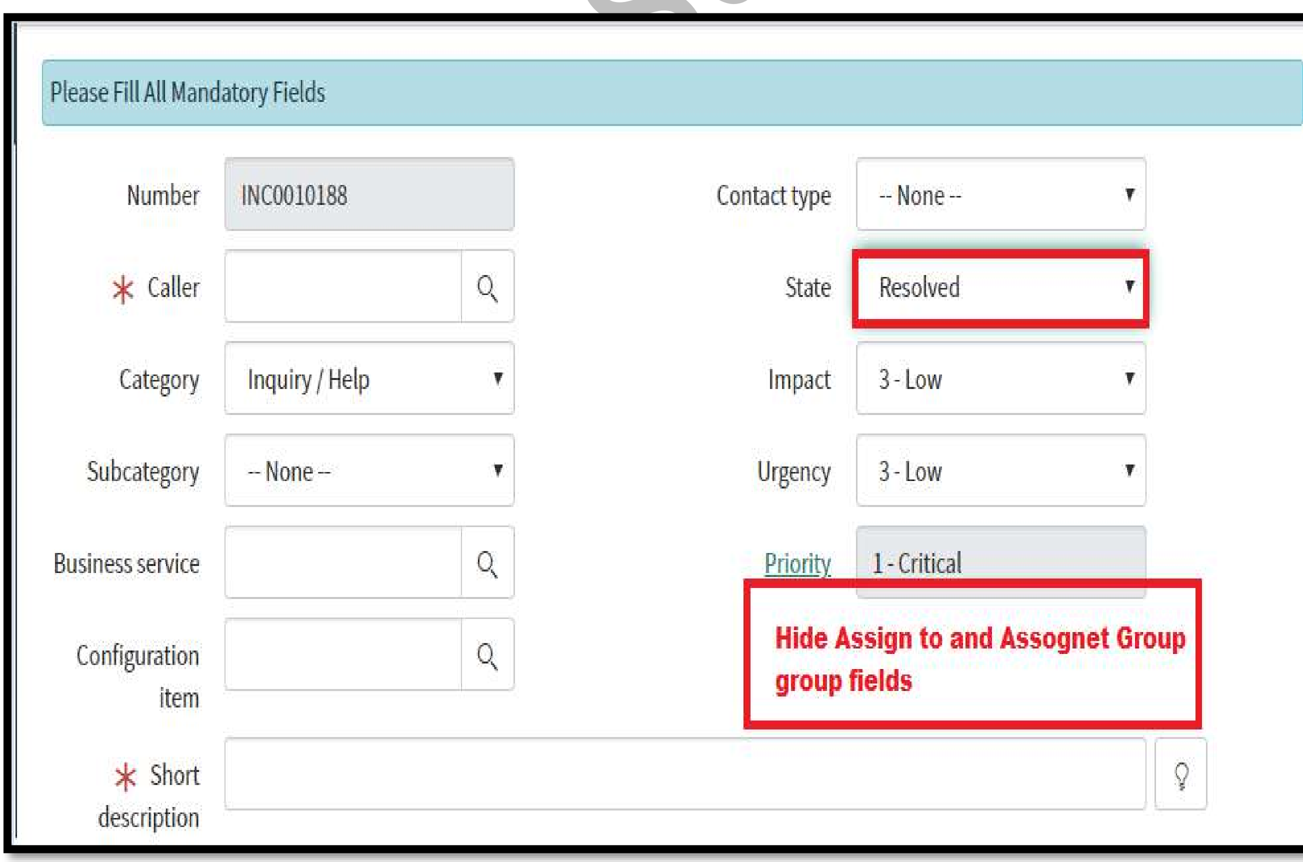


16. Assignment group Visible **False**
17. Click on **Submit**





18. Check in **Incident records** how it is working



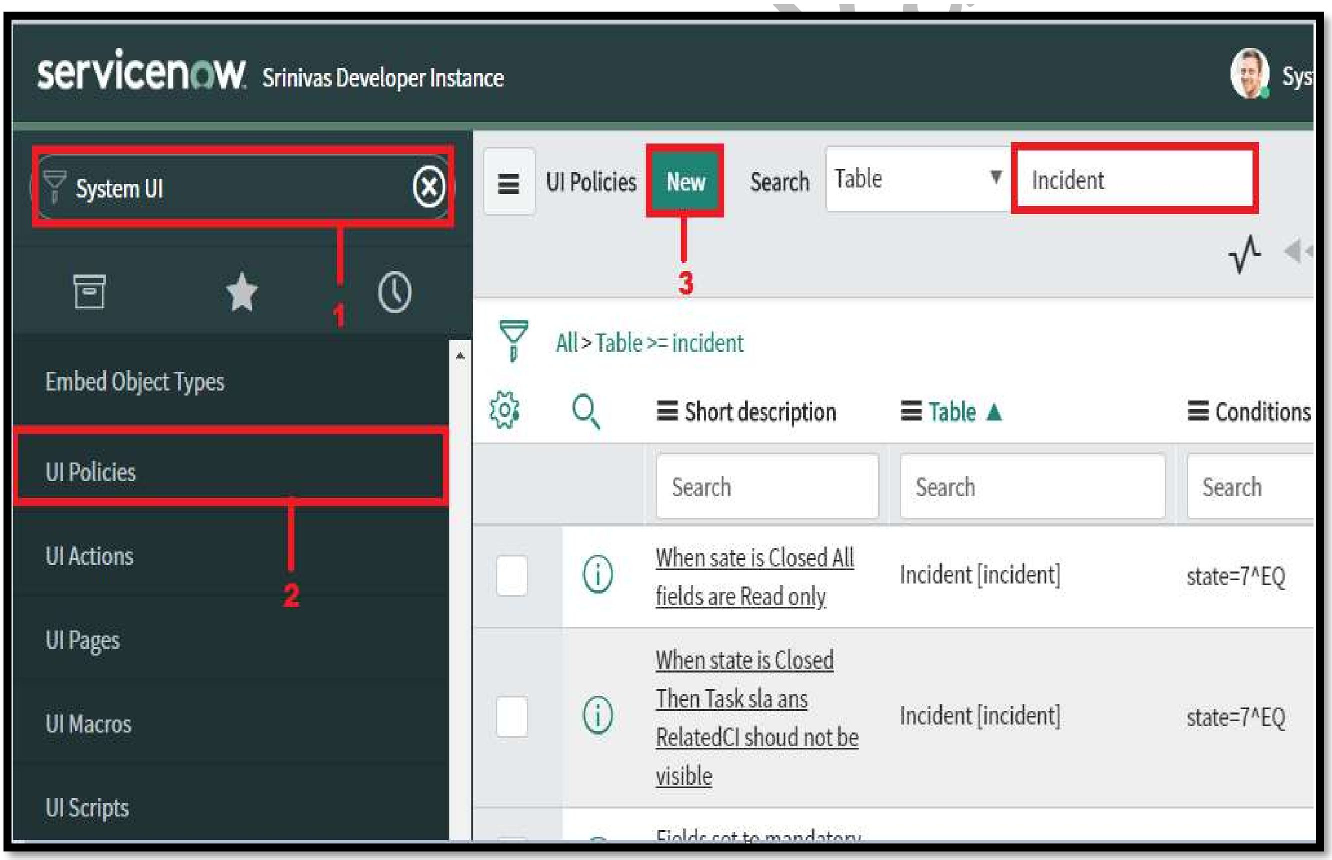
Exercise: 4 About This Requirement

When the Incident state is **Resolved** Then **Resolution Code** and **Resolution notes** fields are **Mandatory**

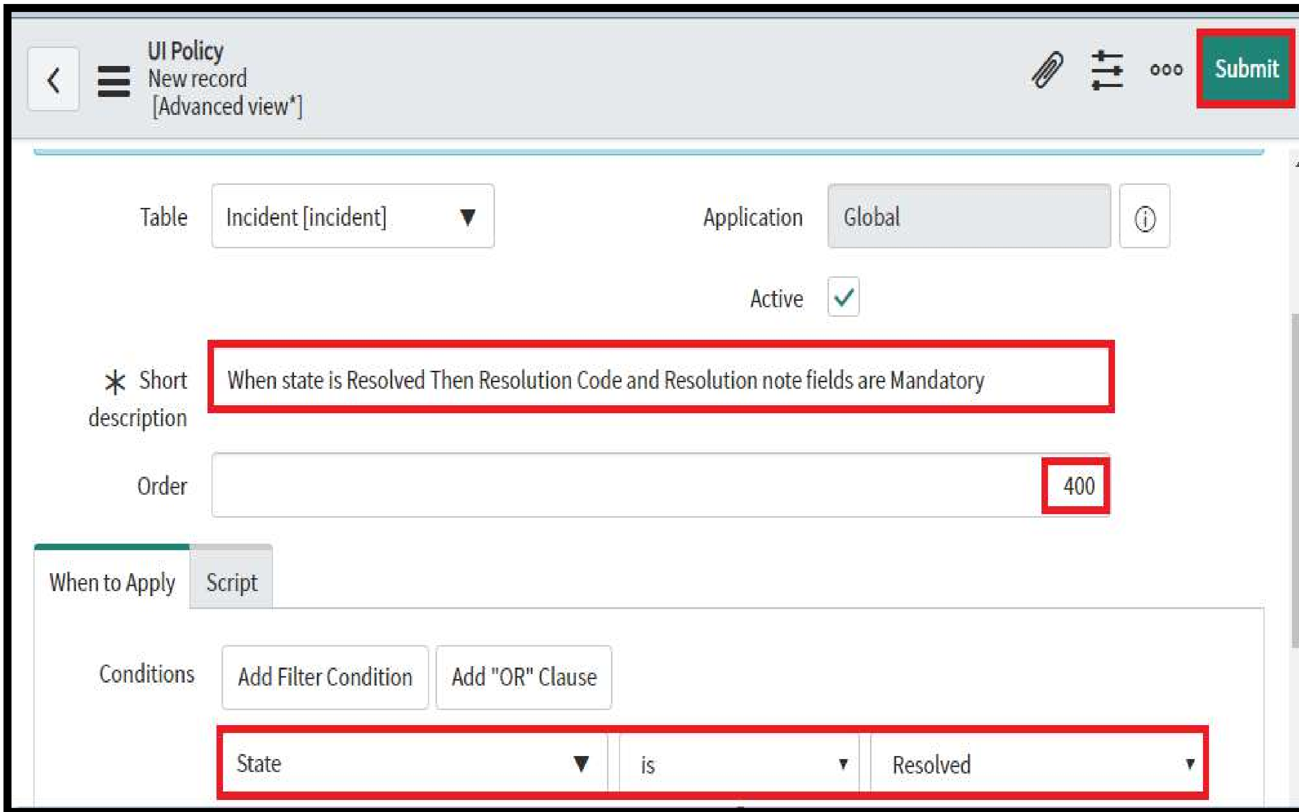
- **Condition** : **[State] [Is] [Resolved]**
- **Action** : **Resolution Code** Mandatory **True**
- **Action** : **Resolution Notes** Mandatory **True**

Procedure

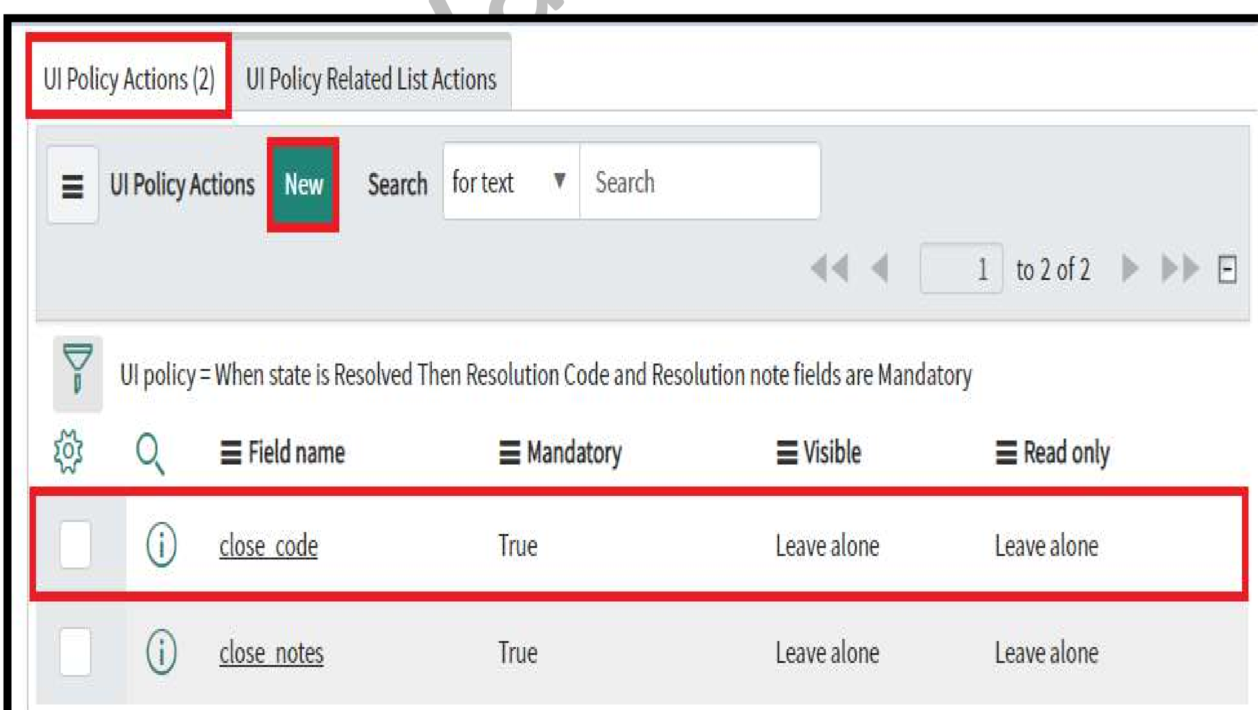
1. Navigate to **System UI** > **UI Policies**.
2. Click **New**.
3. Fill the form with below details



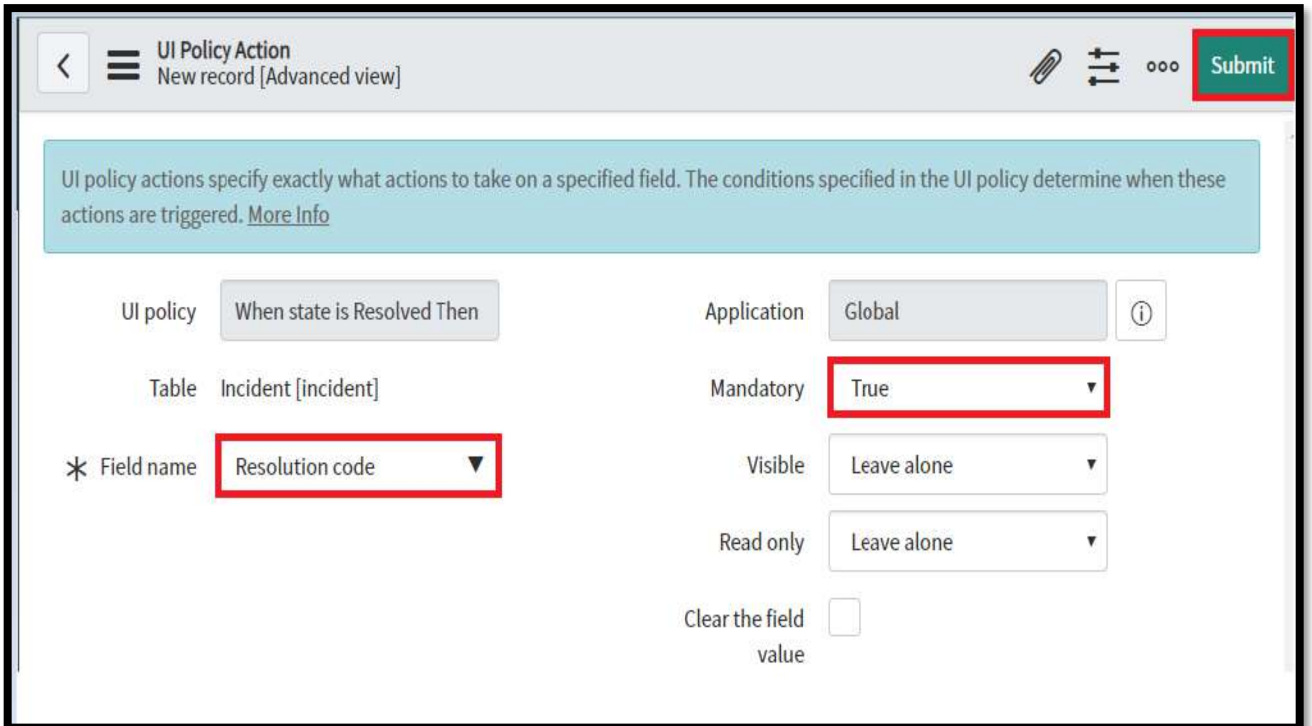
4. **Table:** Incident
5. **Conditions:** state is **Resolved**
6. **Order** :**400**
7. **Short Description:** When Incident state is **Resolved** Then **Resolution Code** and **Resolution Notes** fields are **Mandatory**
8. Click on **Submit**



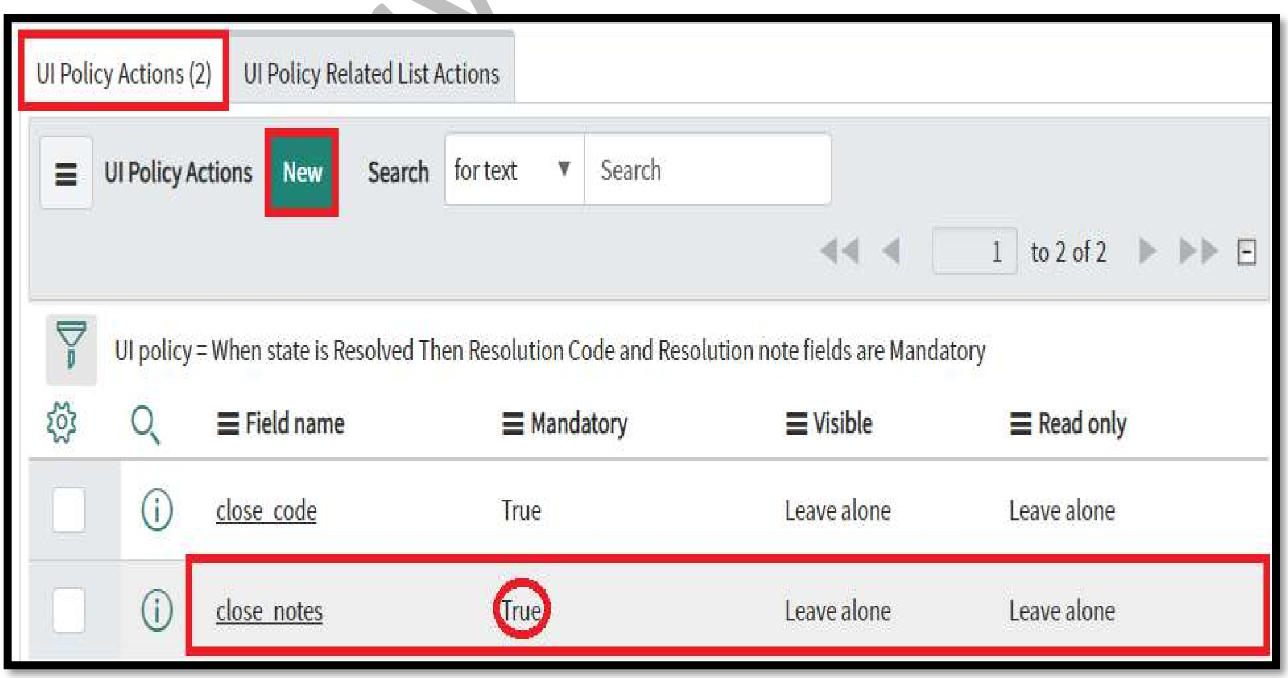
9. Again open the same **UI policy record** and apply **UI policy** action
10. Go to **UI Policy Action** tab from **Related List**



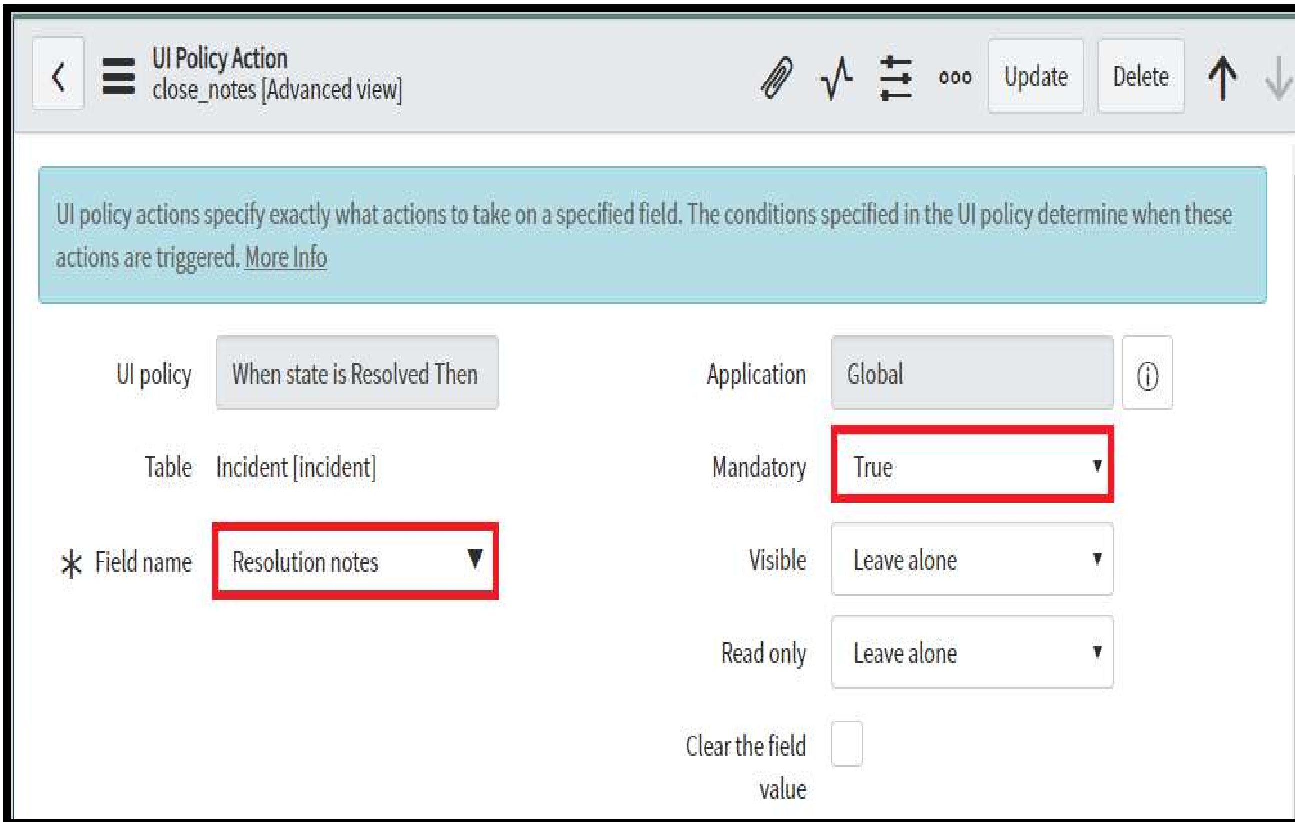
11. Click on UI Policy Action **New**
12. Set **Resolution Code** mandatory **true**
13. Click on **Submit**



14. Navigate to **UI Policy Action** tab from **Related List**



15. Click on UI Policy Action **New**
16. Set **Resolution Notes** Mandatory **True**
17. Click on **Submit**



18. Navigate > incident record to see how it is working

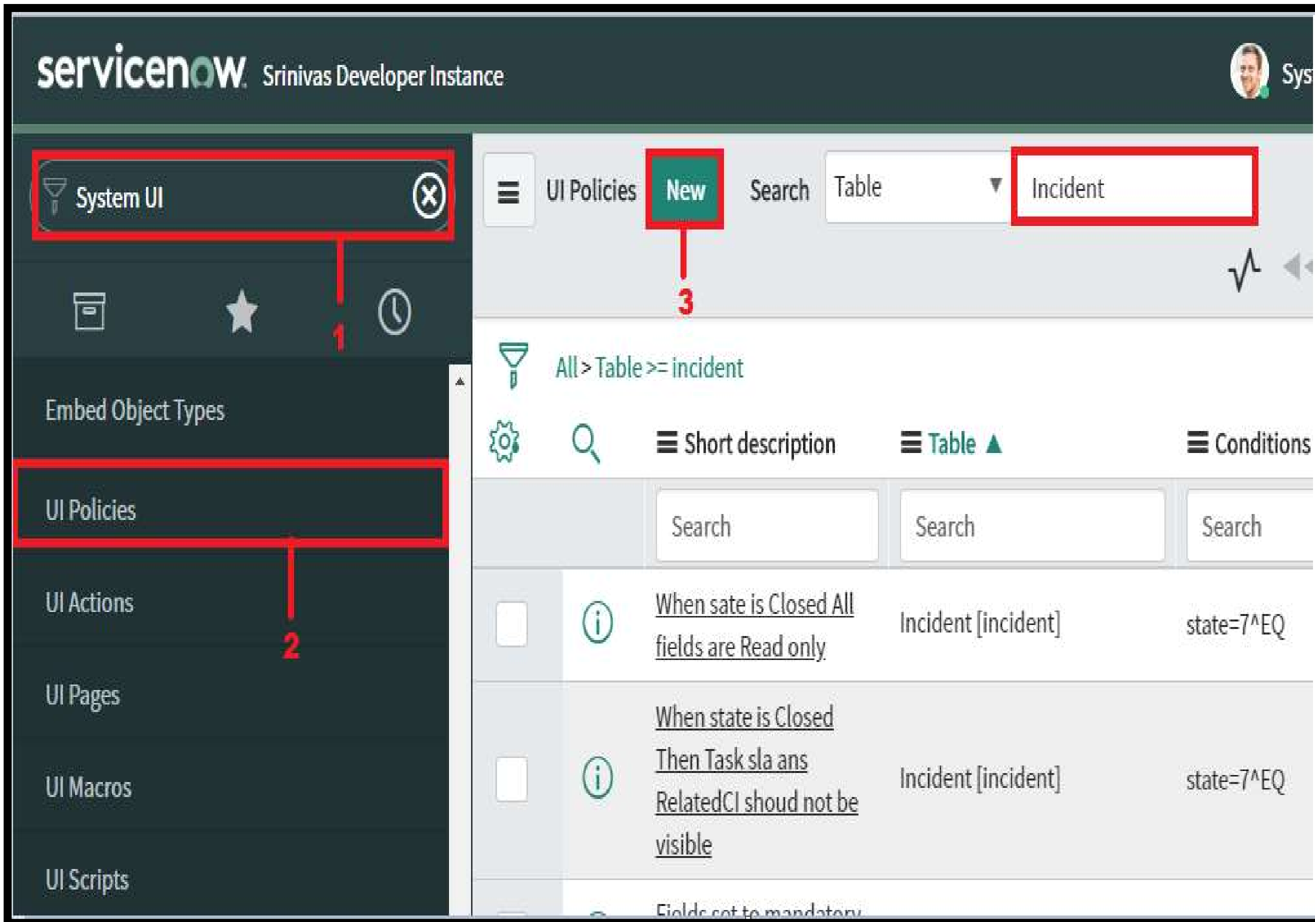
Exercise: 5 About This Requirement

When the Incident **Category** value is **Inquiry/Help** then Subcategory field should not be visible

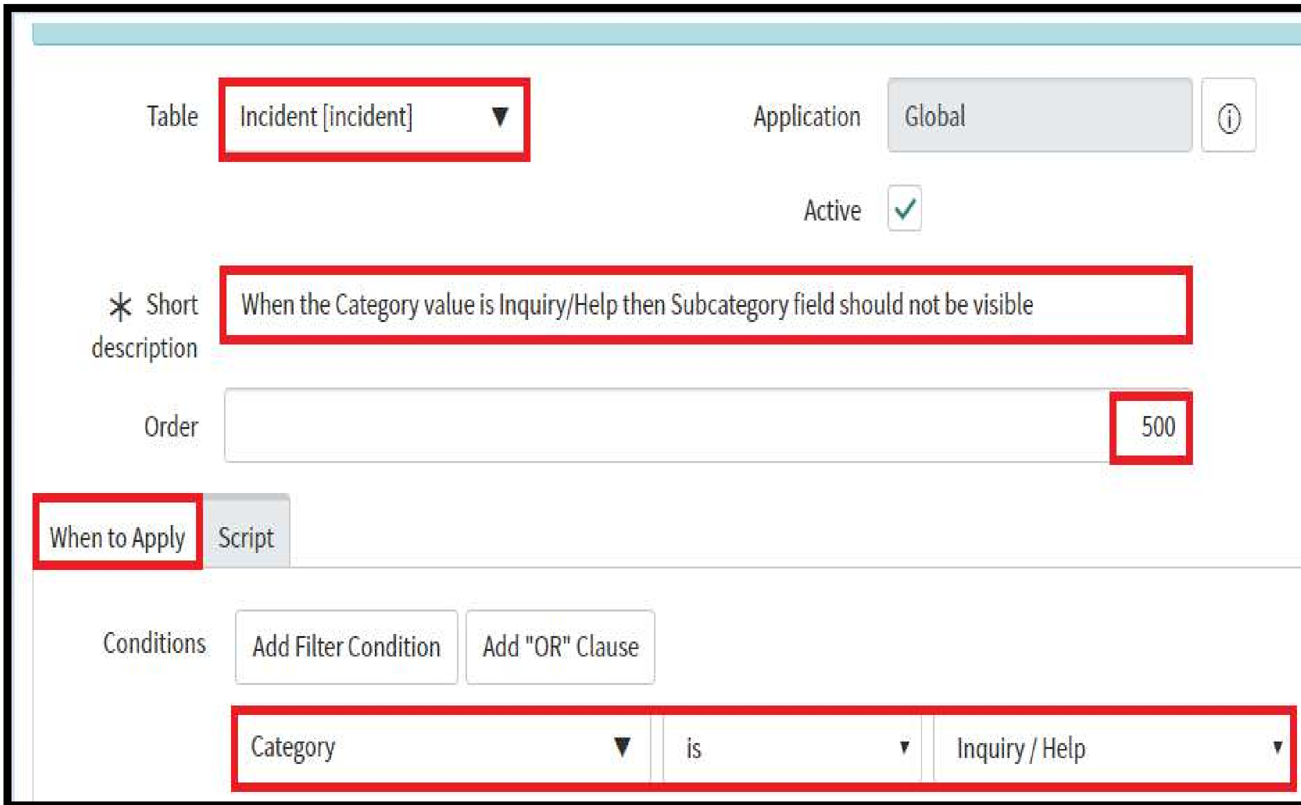
- **Condition** : **[Category] [Is] [Inquiry/Help]**
- **Action** : **Subcategory** Visible **False**

Procedure

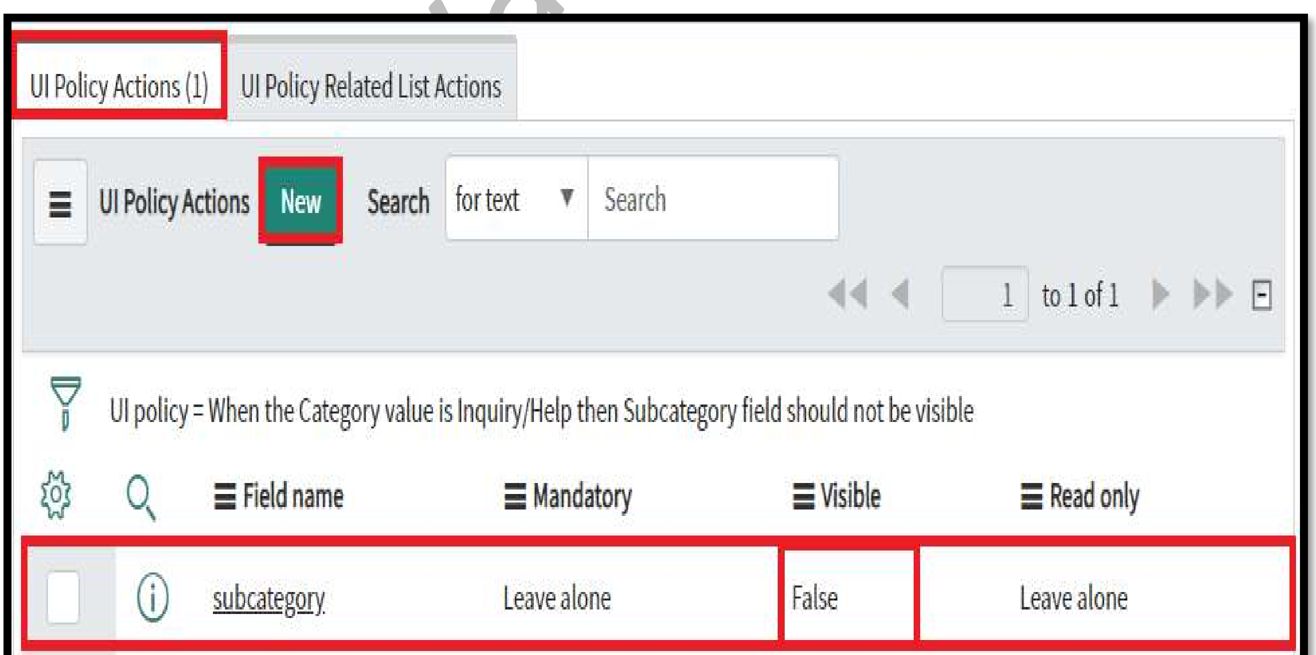
1. Navigate to **System UI** > **UI Policies**.
2. Click **New**.
3. Fill the form with below details



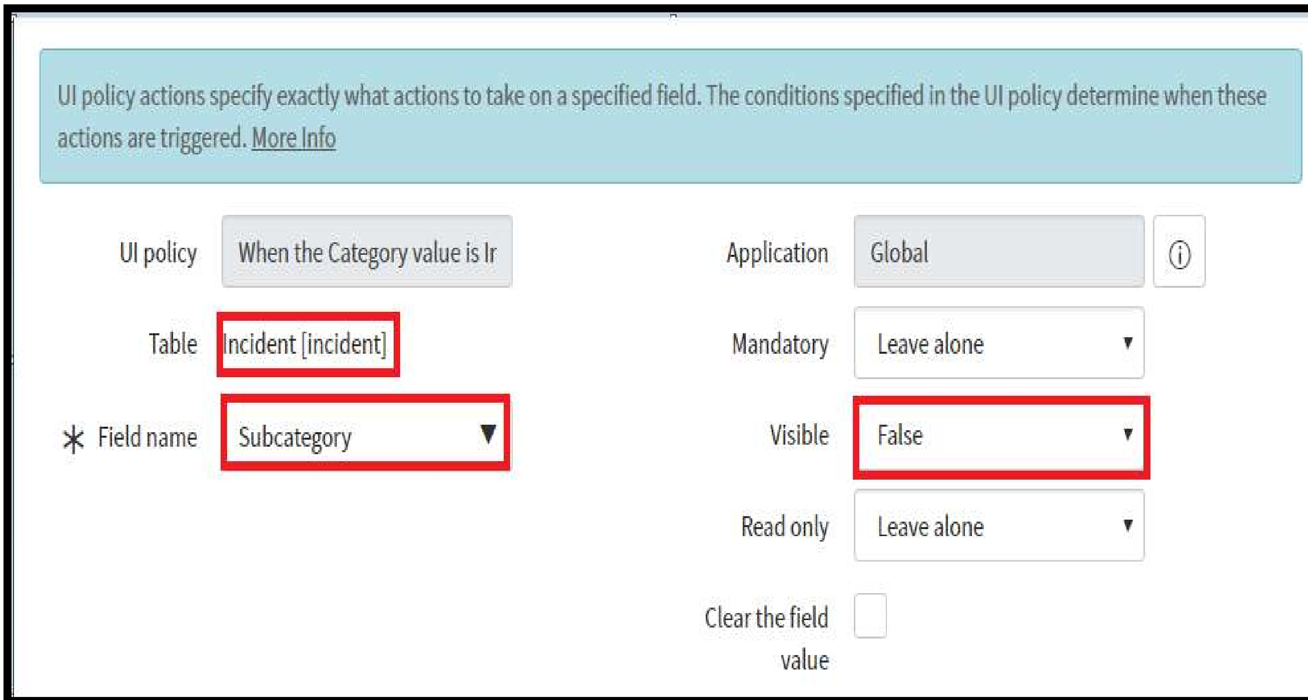
4. **Table:** Incident
5. **Condition: Category** is **Inquiry/Help**
6. **Order** :**500**
7. **Short Description:** When the Incident **Category** value is **Inquiry/Help** then Subcategory field should not be visible
8. Click on **Submit**



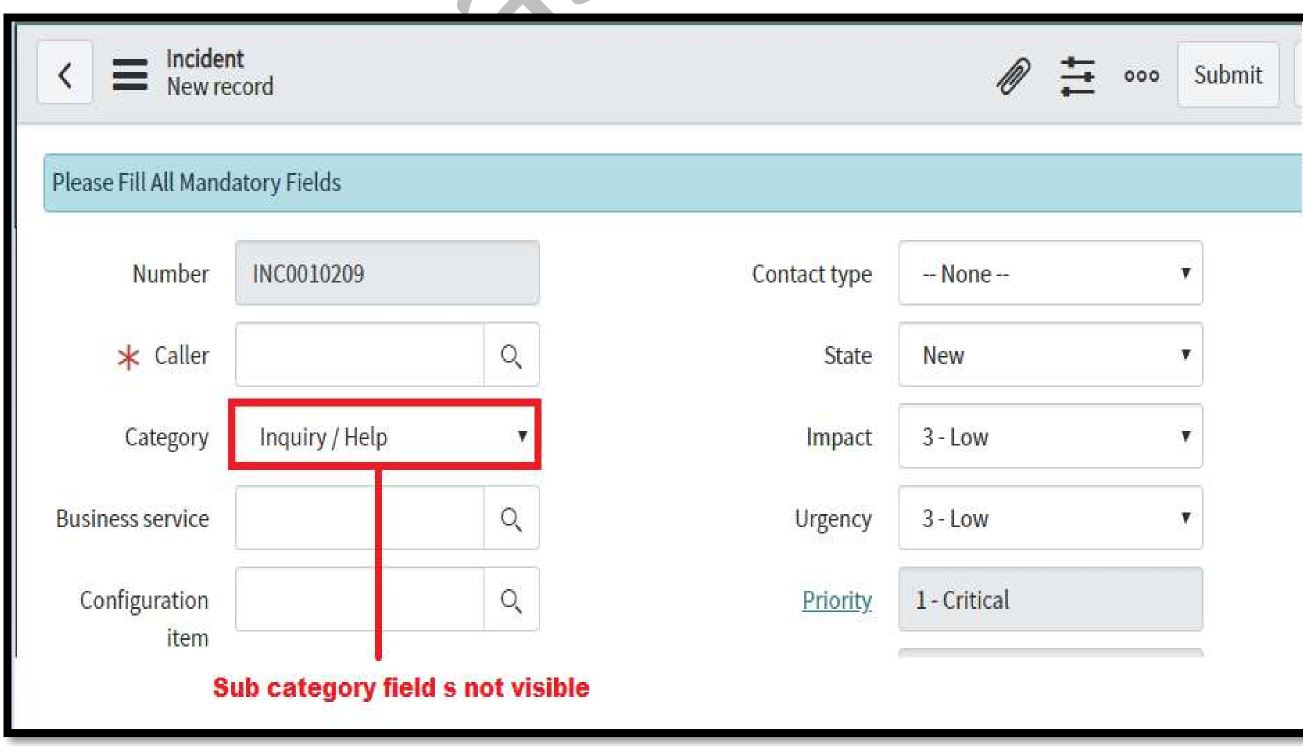
9. Again open the same **UI policy record** and apply **UI policy** action
10. Go to **UI Policy Action** tab from **Related List**



11. Click on UI Policy Action **New**
12. Set **Subcategory** Visible **False**
13. Click on **Submit**



14. Navigate > incident record to see how it is working



Exercise: 6 About This Requirement

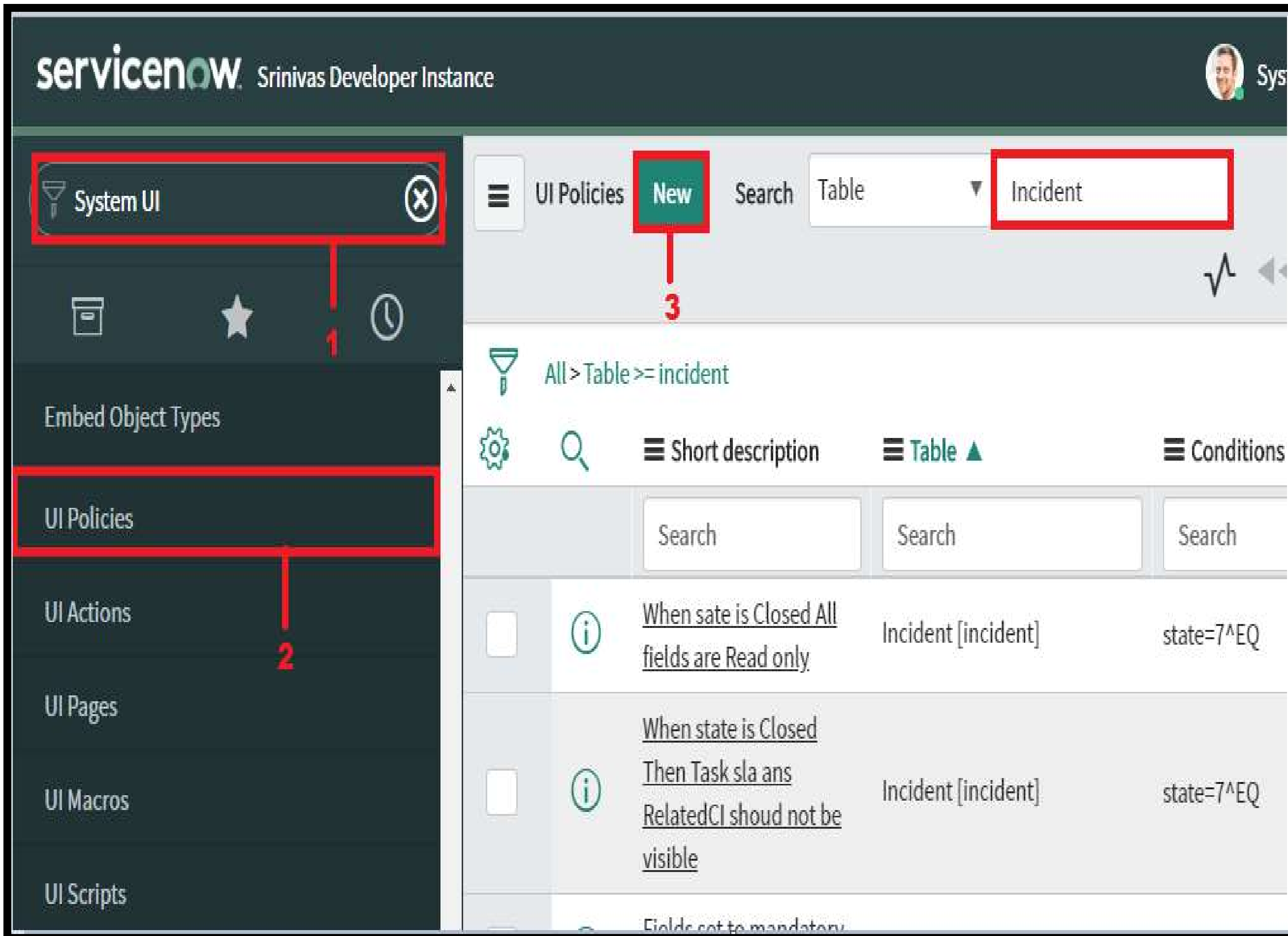
When the Incident **State** is **Closed or Cancelled** Then Number, Caller, Category, Impact, Urgency, State, Work notes, Closed Code, Closed Notes, Assign to, Assignment Group, Comments, Knowledge Configuration Item, Short Description Fields are **Read Only**

- **Condition** : **[State]** [Is] **[Closed] or**
- **Condition** : **[State]** [Is] **[Cancelled]**

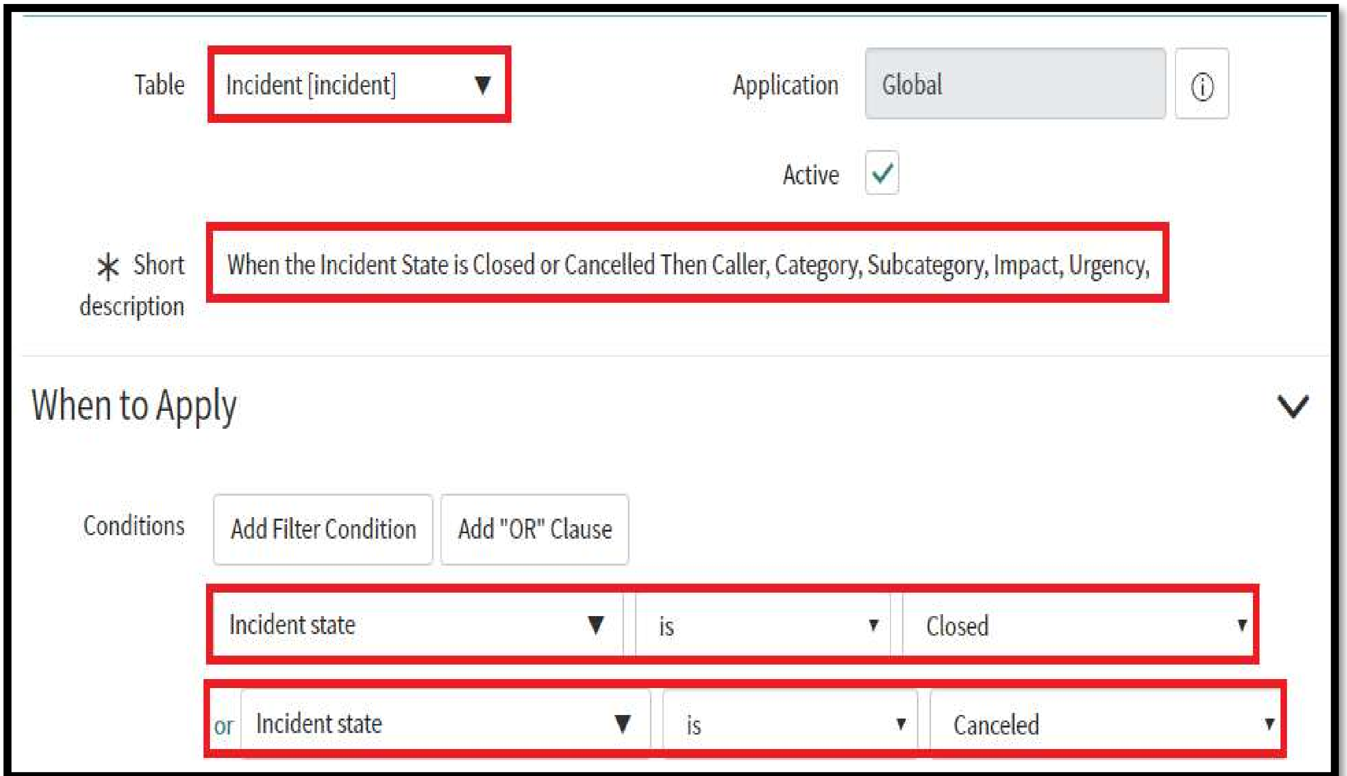
- **Action** : **Caller** Read Only **True**
- **Action** : **Category** Read Only **True**
- **Action** : **State** Read Only **True**
- **Action** : **Impact** Read Only **True**
- **Action** : **Urgency** Read Only **True**
- **Action** : **Assigned to** Read Only **True**
- **Action** : **Assignment Group** Read Only **True**
- **Action** : **Configuration Item** Read Only **True**
- **Action** : **Short Description** Read Only **True**
- **Action** : **Work Notes** Read Only **True**
- **Action** : **Closed Notes** Read Only **True**
- **Action** : **Knowledge** Read Only **True**
- **Action** : **Closed Code** Read Only **True**

Procedure

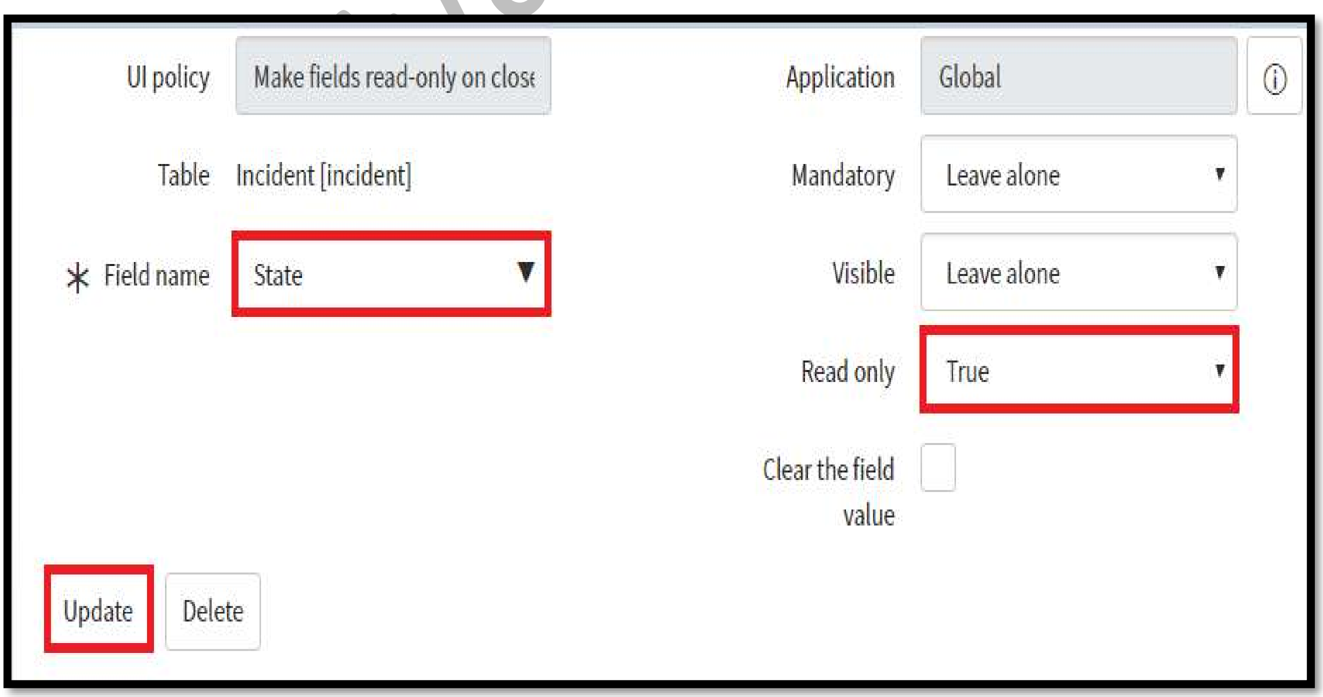
1. Navigate to **System UI** > **UI Policies**.
2. Click **New**.
3. Fill the form with below details



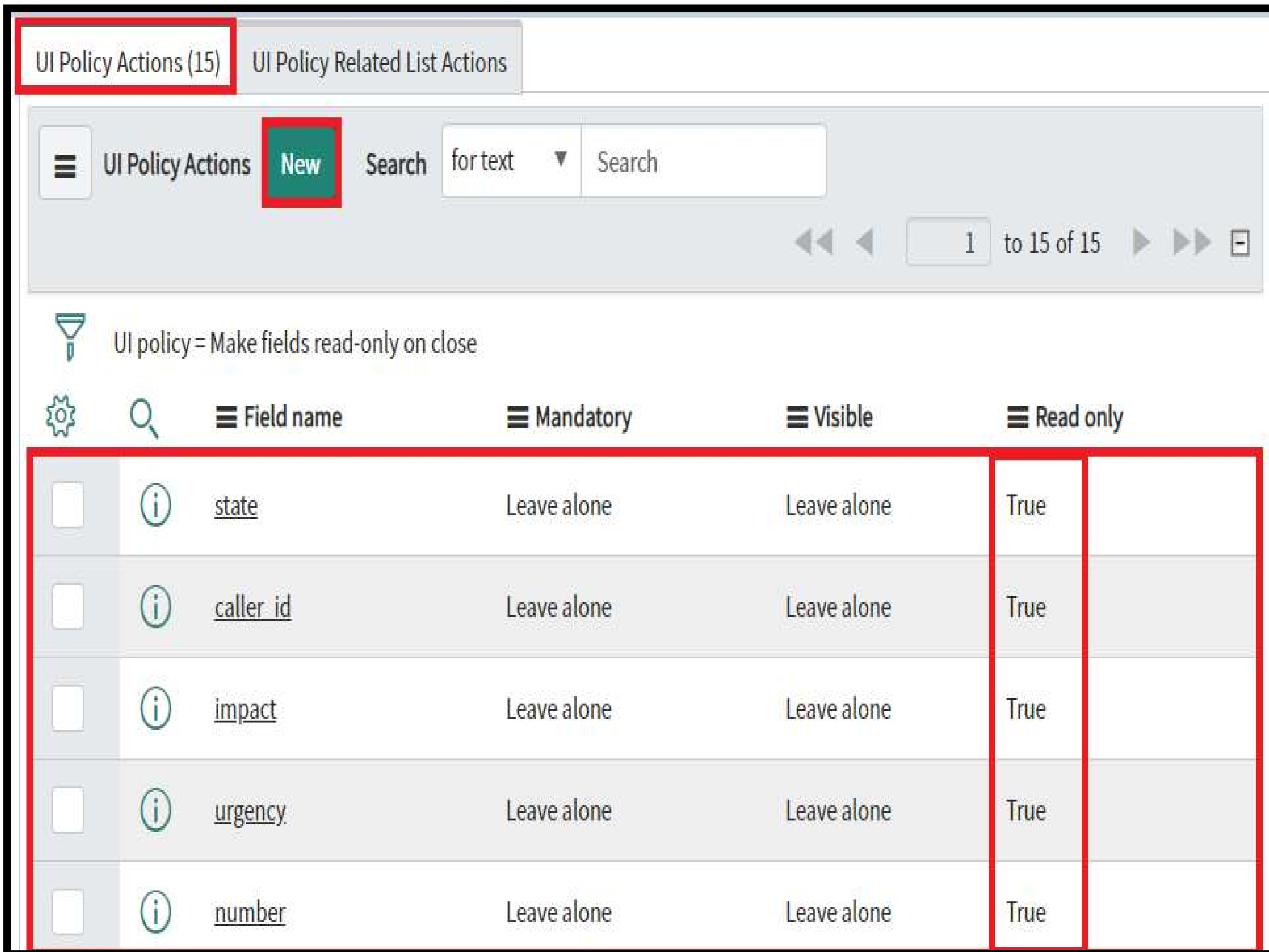
4. **Table:** Incident
5. **Condition: State** is **Closed (or)**
6. **Condition: State** is **Cancelled**
7. **Order** :**600**
8. **Short Description:** When the Incident **State** is **Closed or Cancelled** Then Number, Caller, Category, Subcategory, Impact, Urgency, State, Work notes, Closed Code, Closed Notes, Assign to, Assignment Group, Comments, Knowledge Configuration Item, Short Description Fields are **Read Only**
9. Click on **Submit**



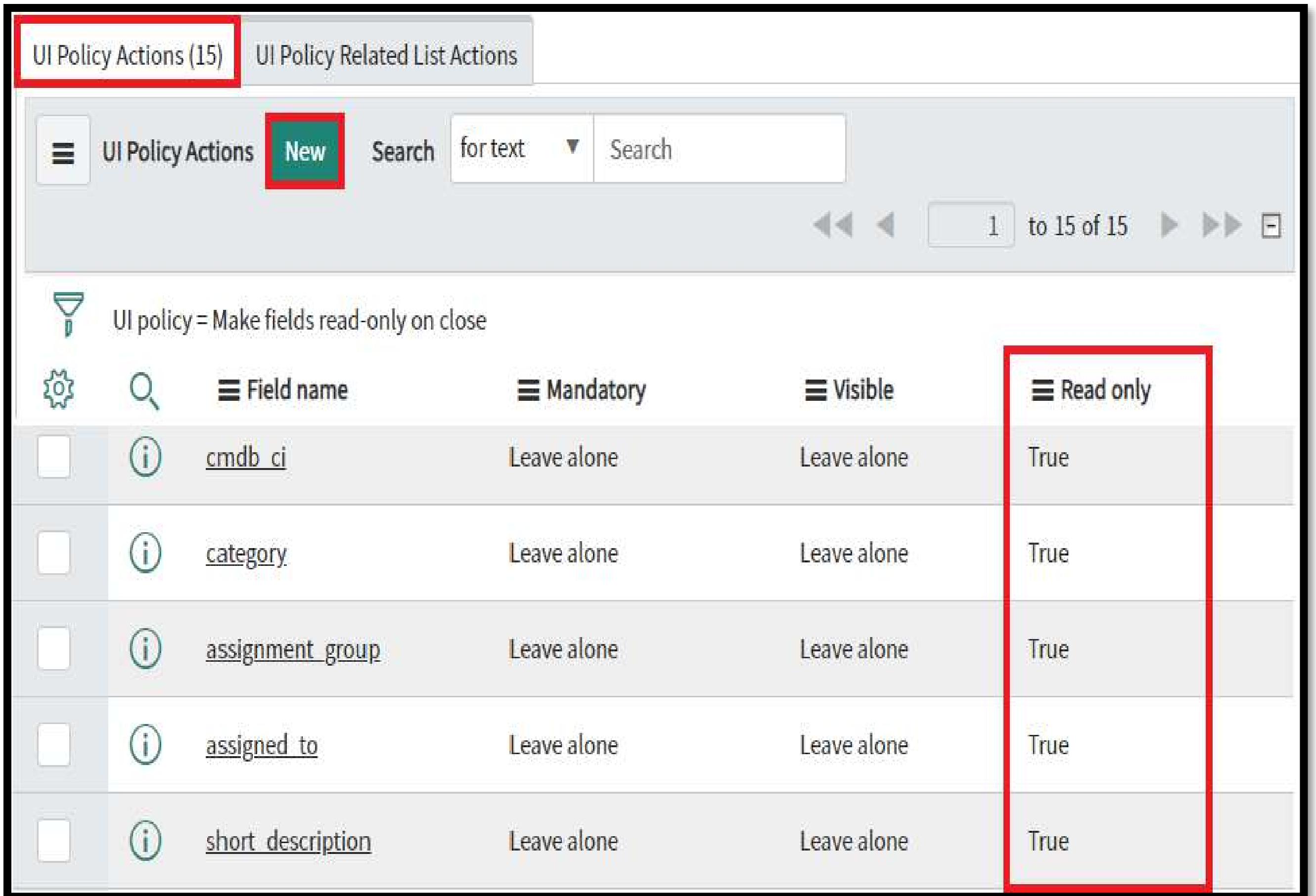
10. Again open the same **UI policy record** and apply **UI policy** action
11. Go to **UI Policy Action** tab from **Related List**
12. Click on UI Policy Action **New**



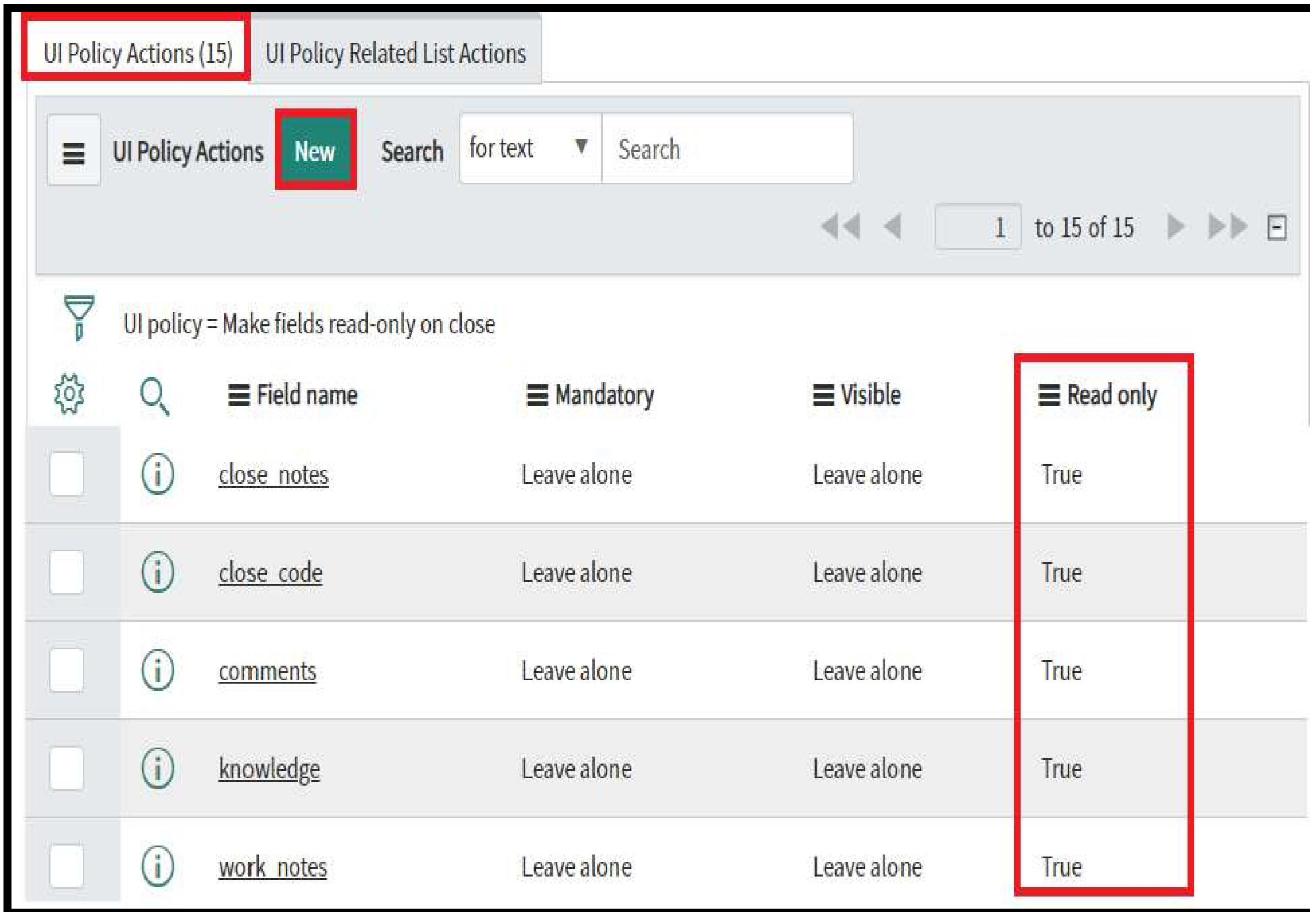
13. Set **State** Read Only **True**
14. Set **Caller** Read Only **True**
15. Set **Impact** Read Only **True**
16. Set **Urgency** Read Only **True**
17. Set **Number** Read Only **True**
18. Click on **Submit**



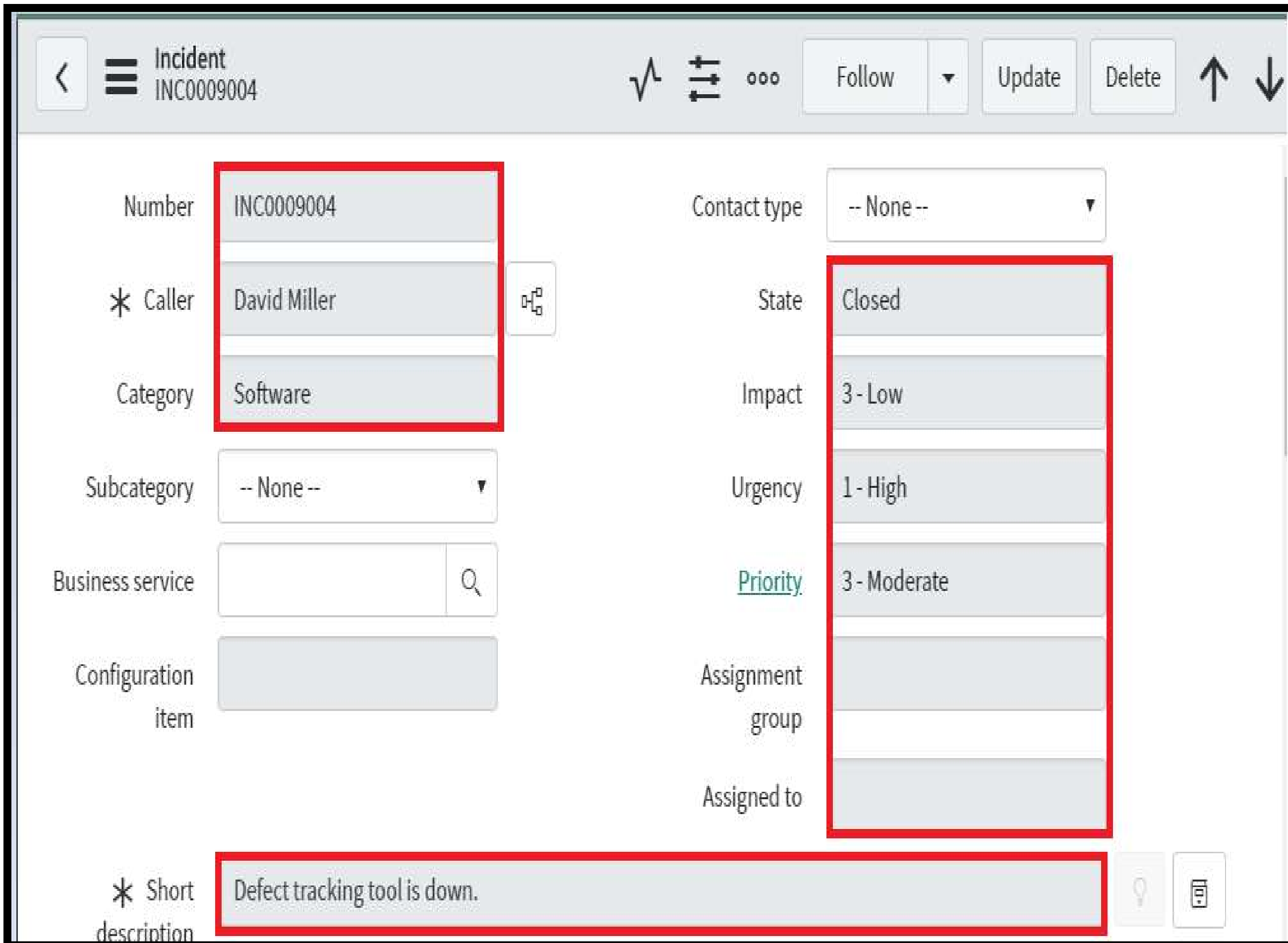
19. Set **State** Read Only **True**
20. Set **Caller** Read Only **True**
21. Set **Impact** Read Only **True**
22. Set **Urgency** Read Only **True**
23. Set **Number** Read Only **True**
24. Click on **Submit**



25. Set **Configuration Item** Read Only **True**
26. Set **Category** Read Only **True**
27. Set **Assignment Group** Read Only **True**
28. Set **Assign to** Read Only **True**
29. Set **Short Description** Read Only **True**



30. Set **Close Notes** Read Only **True**
31. Set **Close Code** Read Only **True**
32. Set **Comments** Read Only **True**
33. Set **Knowledge** Read Only **True**
34. Set **Work Notes** Read Only **True**



35. Navigate > incident record to see how it is working

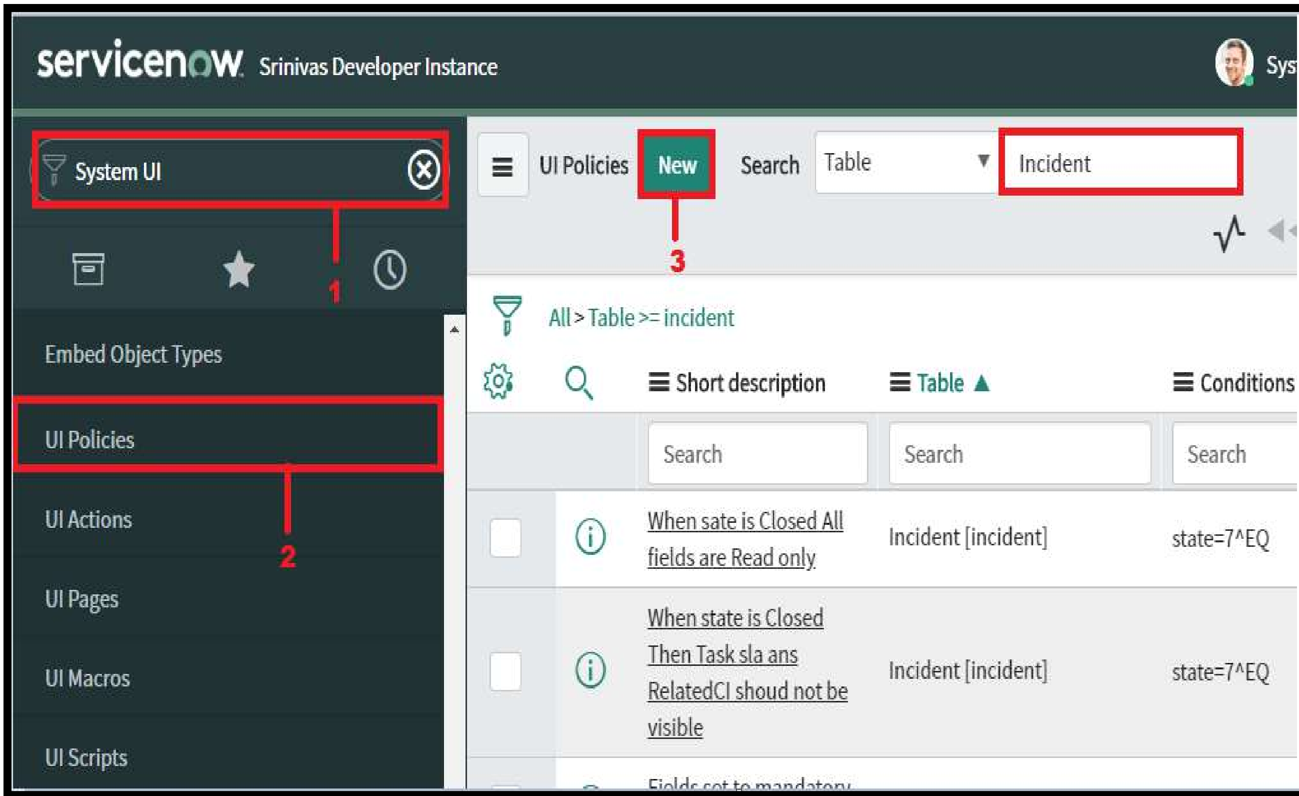
Exercise: 7 About This Requirement

When the Incident **State** is **On-Hold** and **On Hold Reason** is **Awaiting Caller** Then **Comments** field is make **Mandatory**

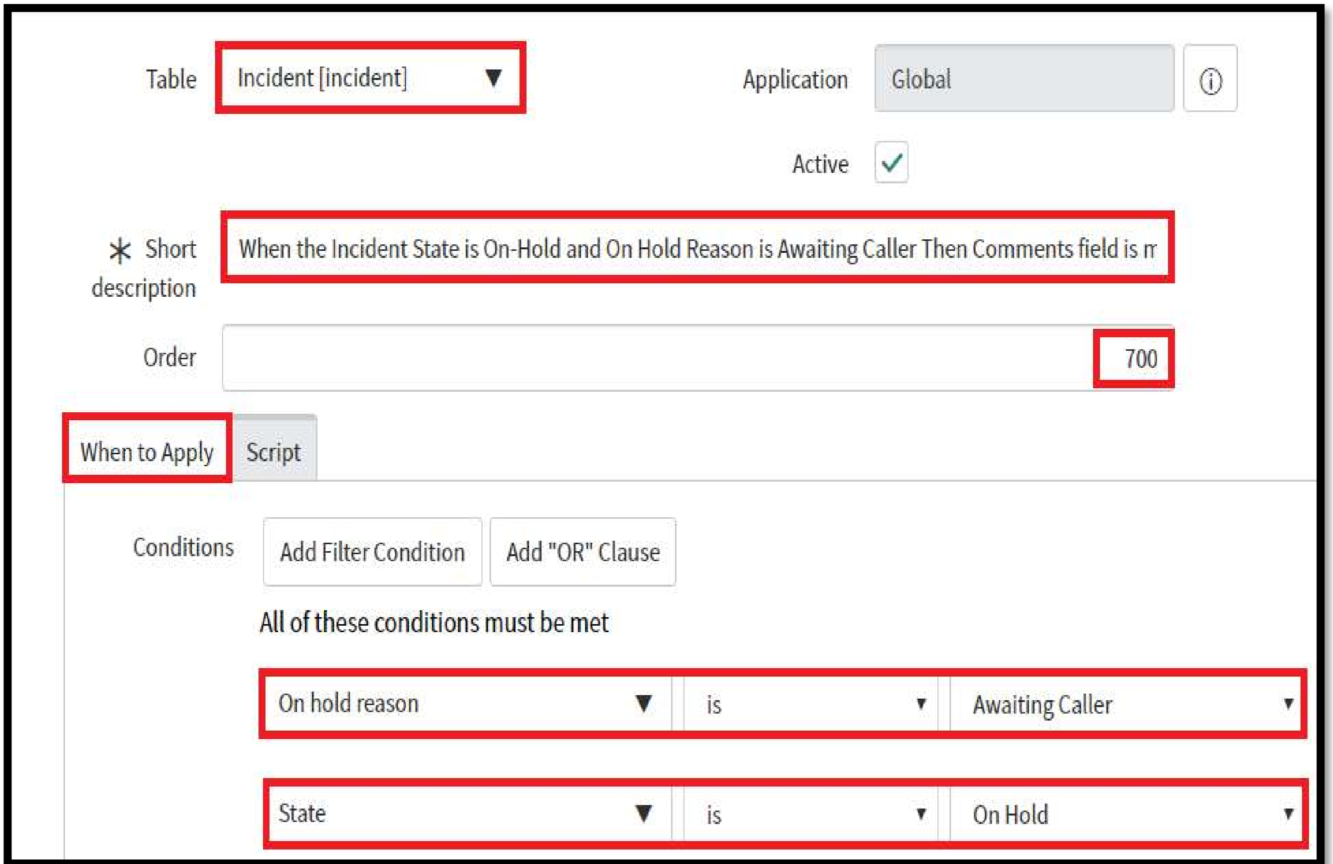
- **Condition** : **[State]** [Is] **[On-Hold] and**
- **Condition** : **[On Hold Reason]** [Is] **[Awaiting Caller]**
- **Action** : **Comments** Mandatory **True**

Procedure

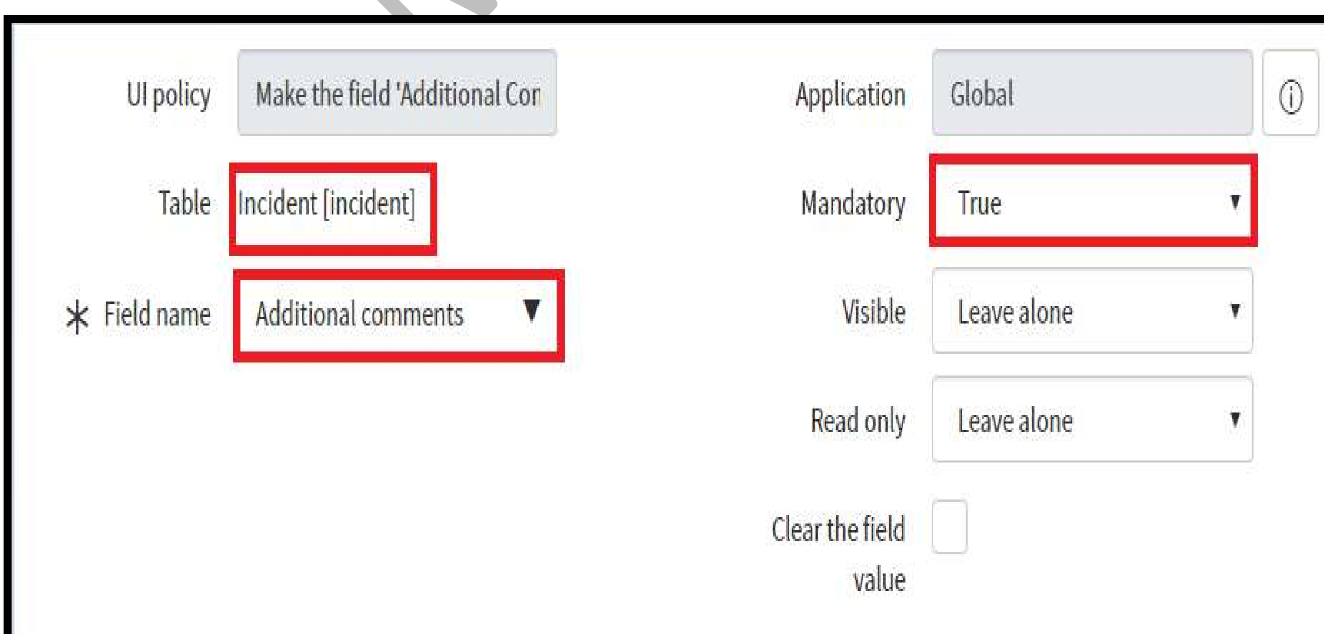
1. Navigate to **System UI** > **UI Policies**.
2. Click **New**.
3. Fill the form with below details



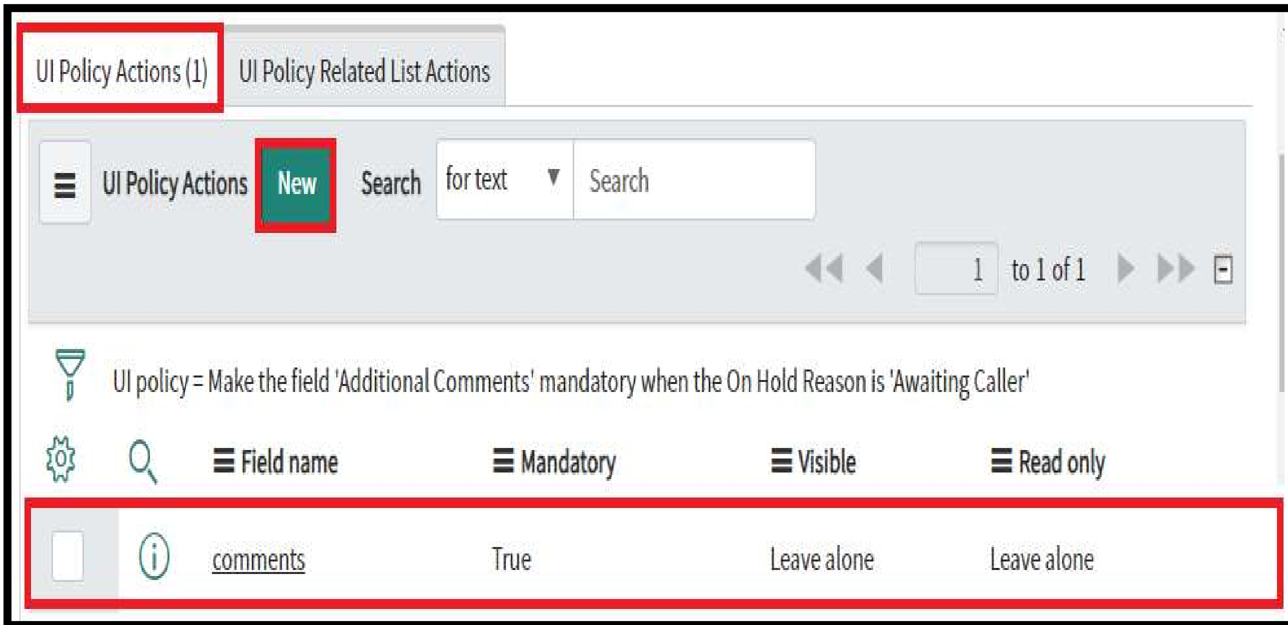
4. **Table:** Incident
5. **Condition: State** is **On Hold (and)**
6. **Condition: On Hold Reason** is **Awaiting Caller**
7. **Order** :**700**
8. **Short Description**: When the Incident **State** is **On-Hold** and **On Hold Reason** is **Awaiting Caller** Then **Comments** field is make **Mandatory**
9. Click on **Submit**



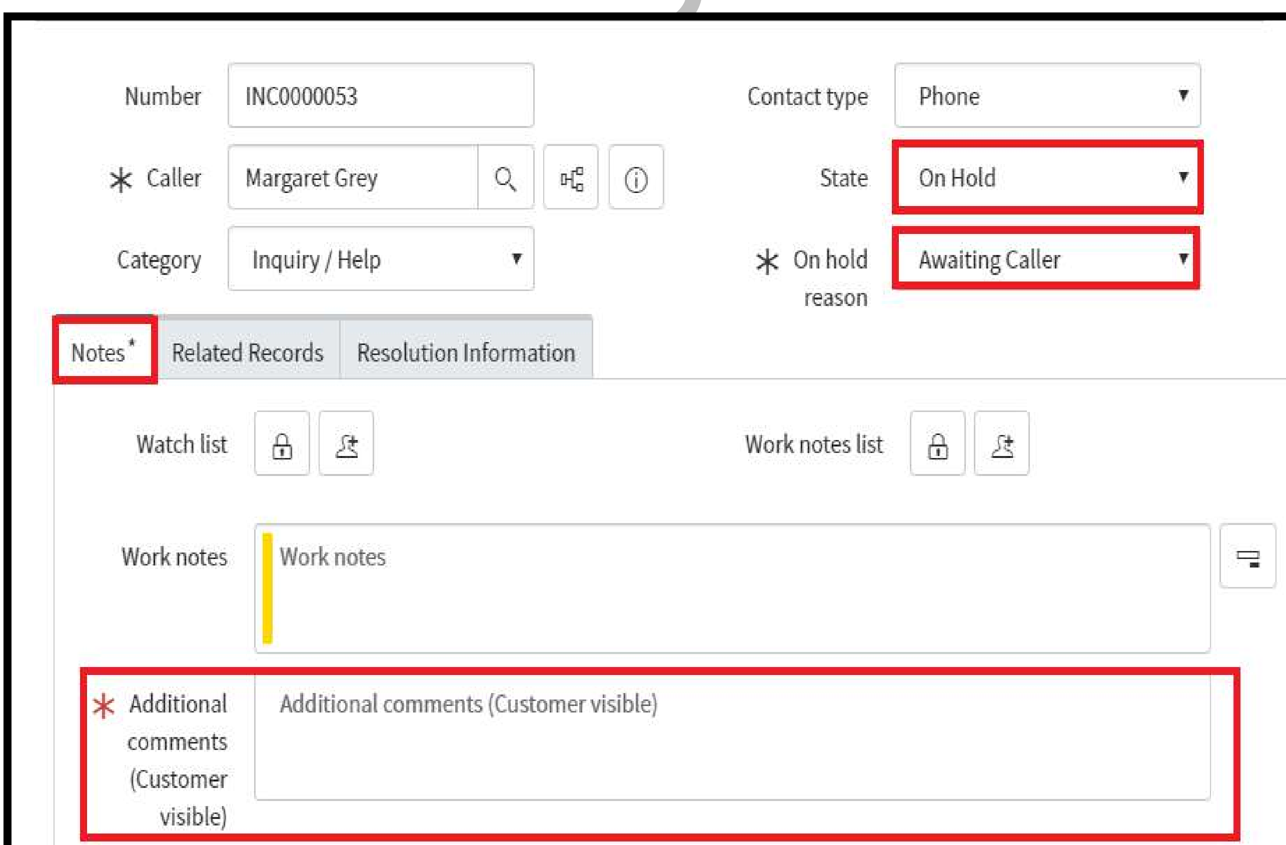
10. Again open the same **UI policy record** and apply **UI policy** action
11. Go to **UI Policy Action** tab from **Related List**
12. Click on UI Policy Action **New**



13. Addition Comments Mandatory **True**
14. Click on **Submit**



15. Navigate > incident record to see how it is working



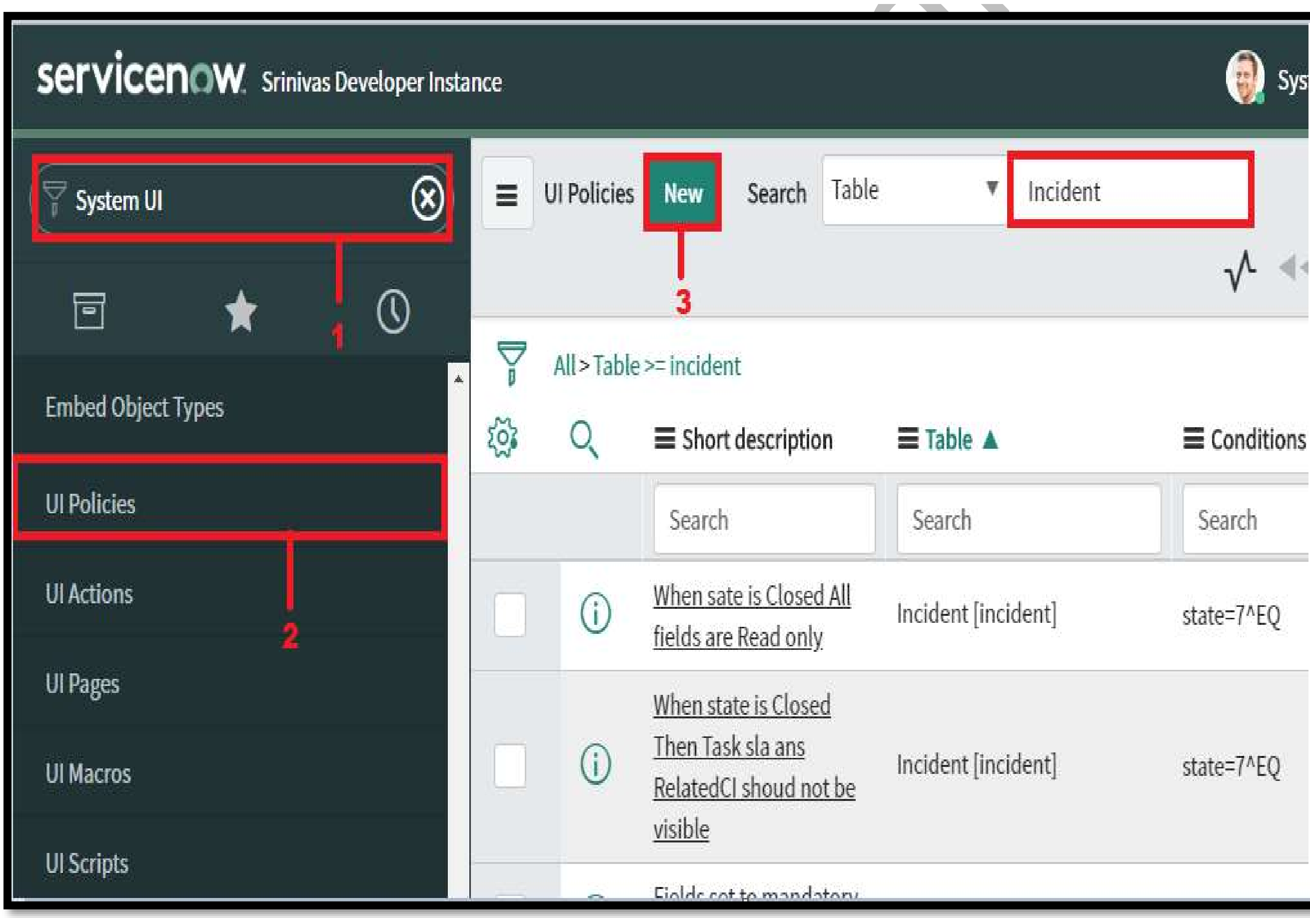
Exercise: 8 About This Requirement

When the Incident **State** is **Closed** Then All the **Related List** should not be Visible in **Incident Form**

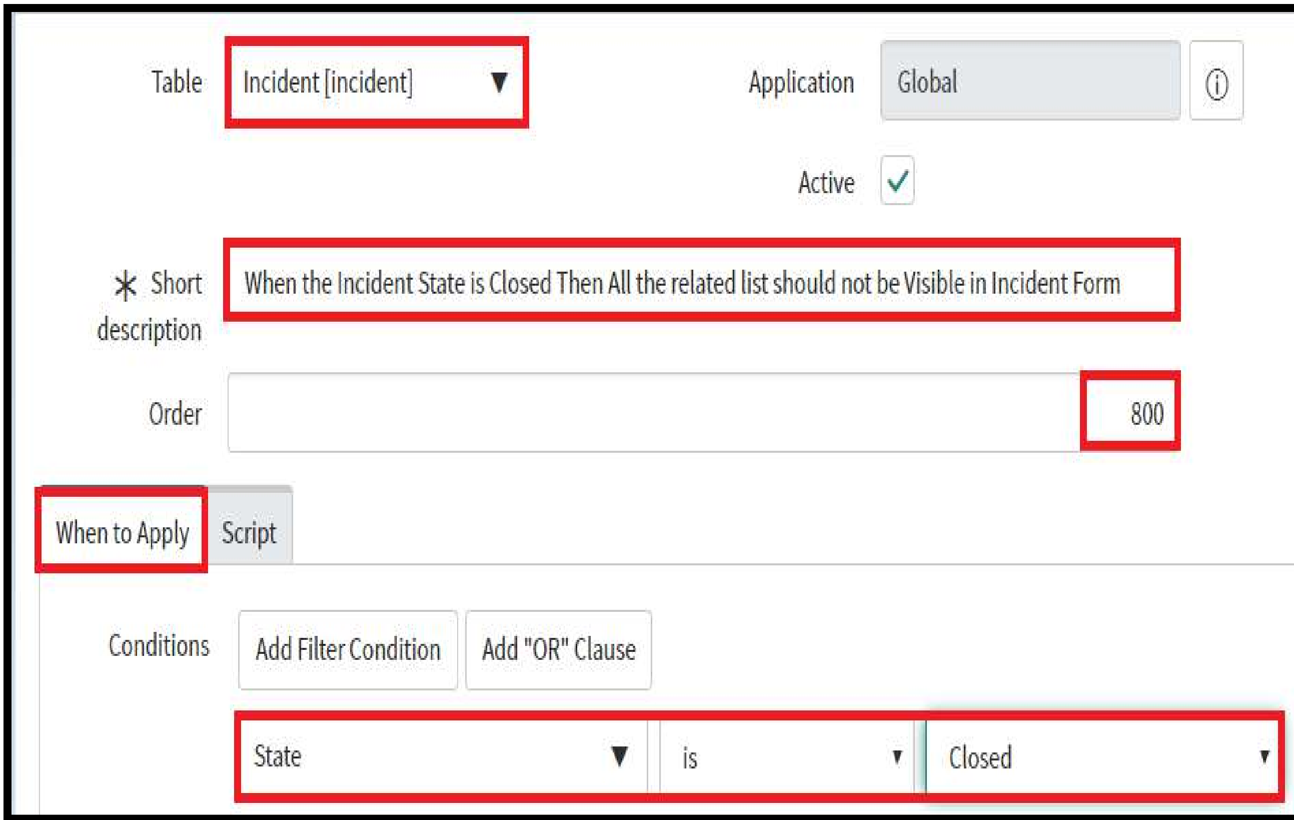
- **Condition** : **[State]** [Is] **[Closed]**
- **Related List Action** : Task SLA, Affected Cis, Impacted Services/Cis, Child Incidents

Procedure

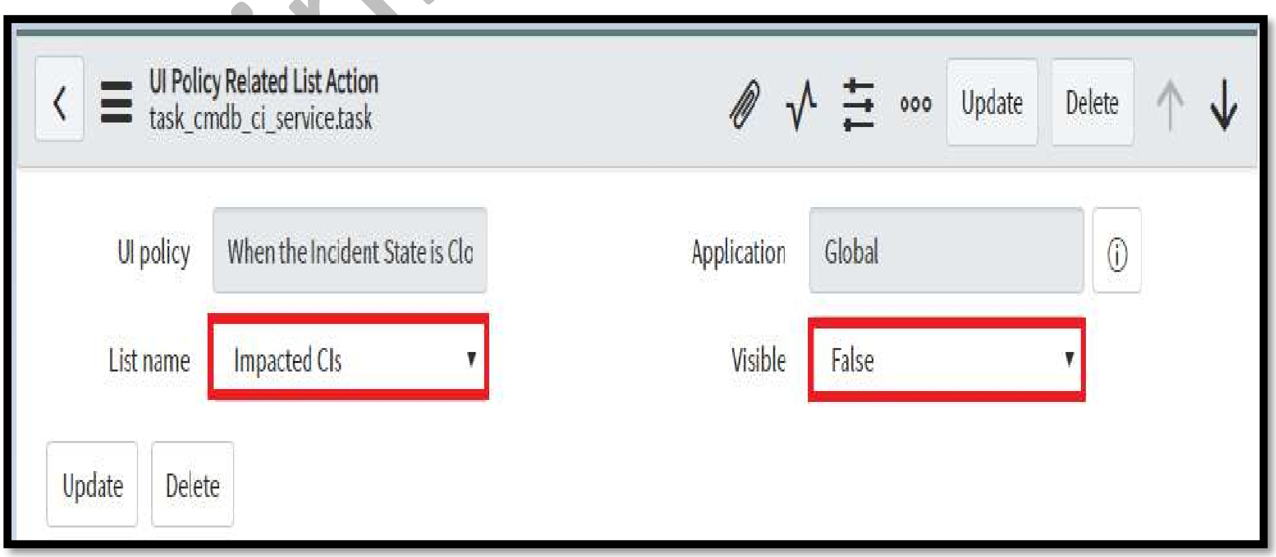
1. Navigate to **System UI** > **UI Policies**.
2. Click **New**.
3. Fill the form with below details



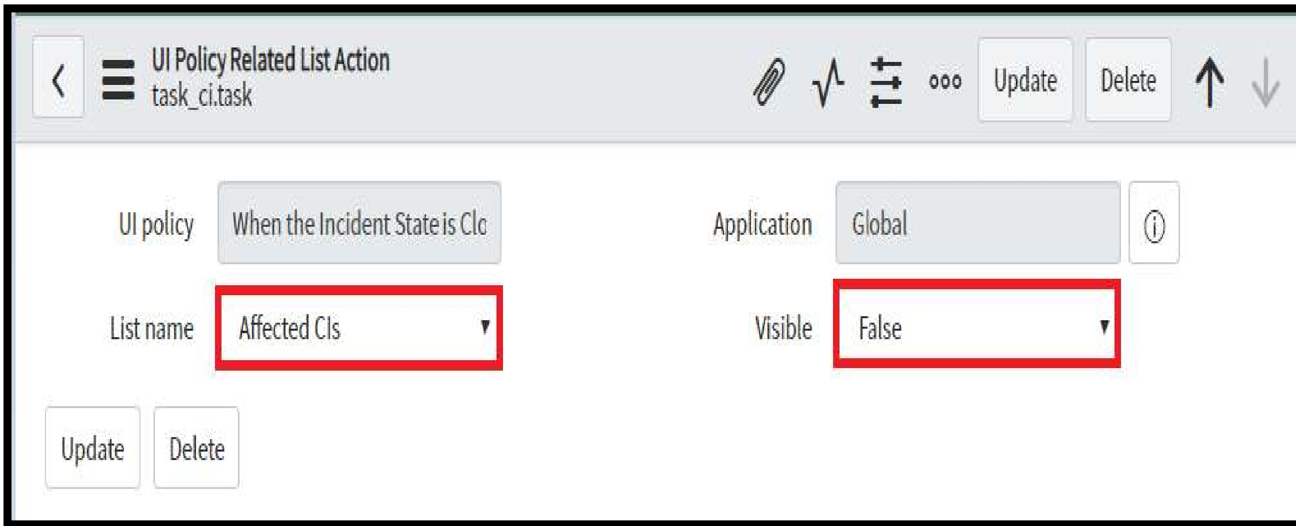
4. **Table:** Incident
5. **Condition: State** is **Closed**
6. **Order** :**800**
7. **Short Description**: When the Incident **State** is **Closed** Then All the related list should not be Visible in **Incident Form**
8. Click on **Submit**



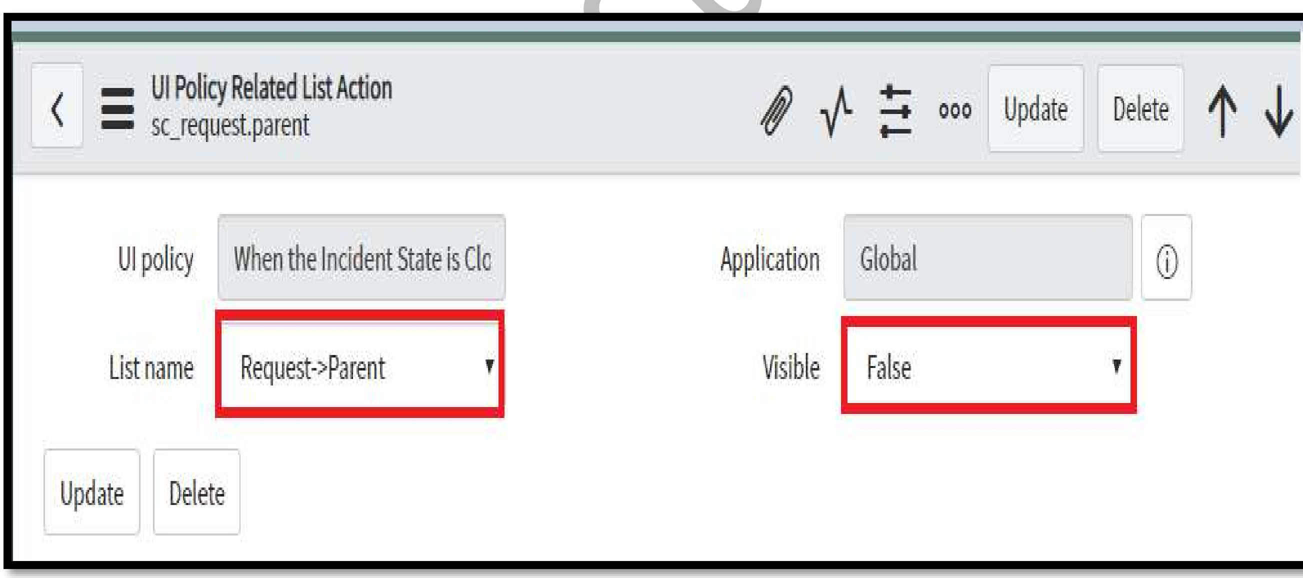
9. Again open the same **UI policy record** and apply **UI Policy Related List** action
10. Go to **UI Policy Related List Action** tab from **Related List**
11. Click on UI Policy Action **New**
12. Set List Name **Impacted Cis** Visible **False**
13. Click on **Submit**

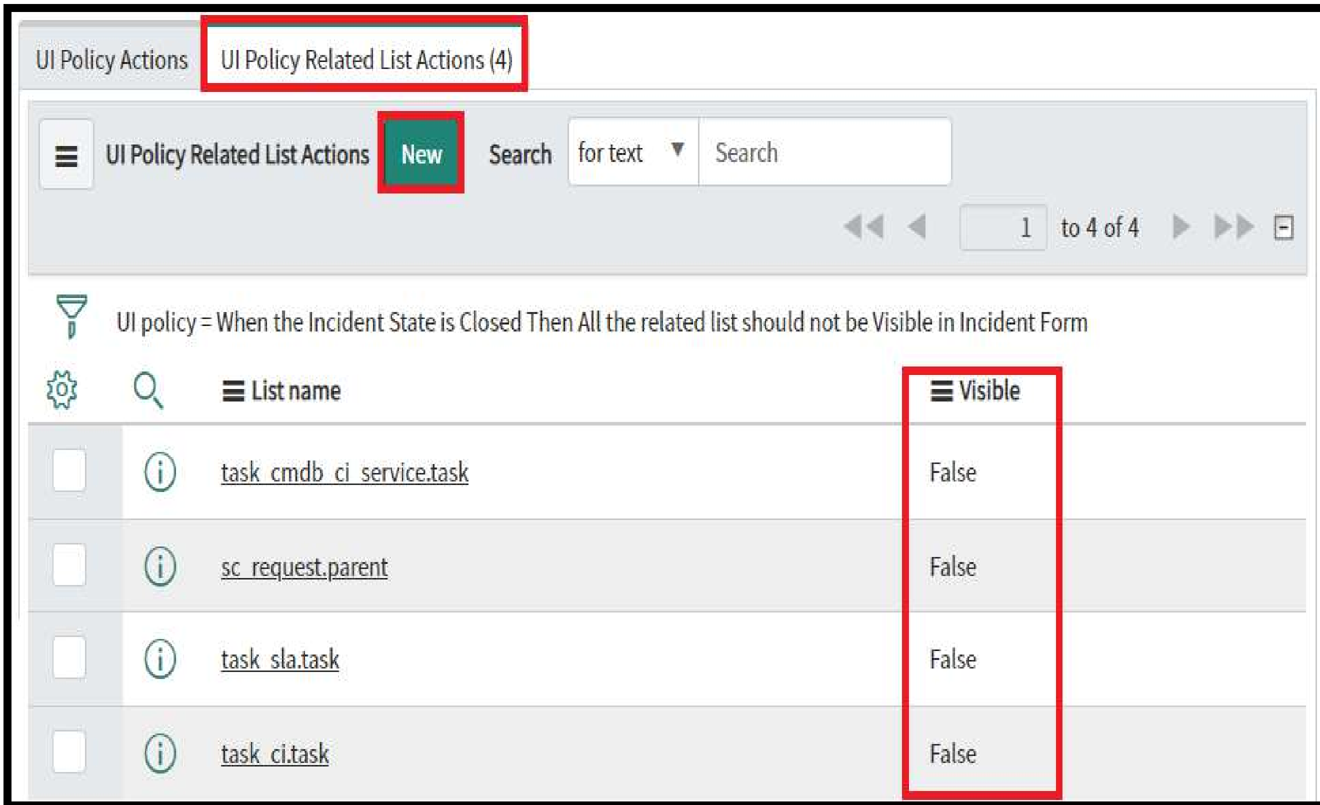


14. Set List Name **Affected CI** Visible **False**
15. Click On **Submit**

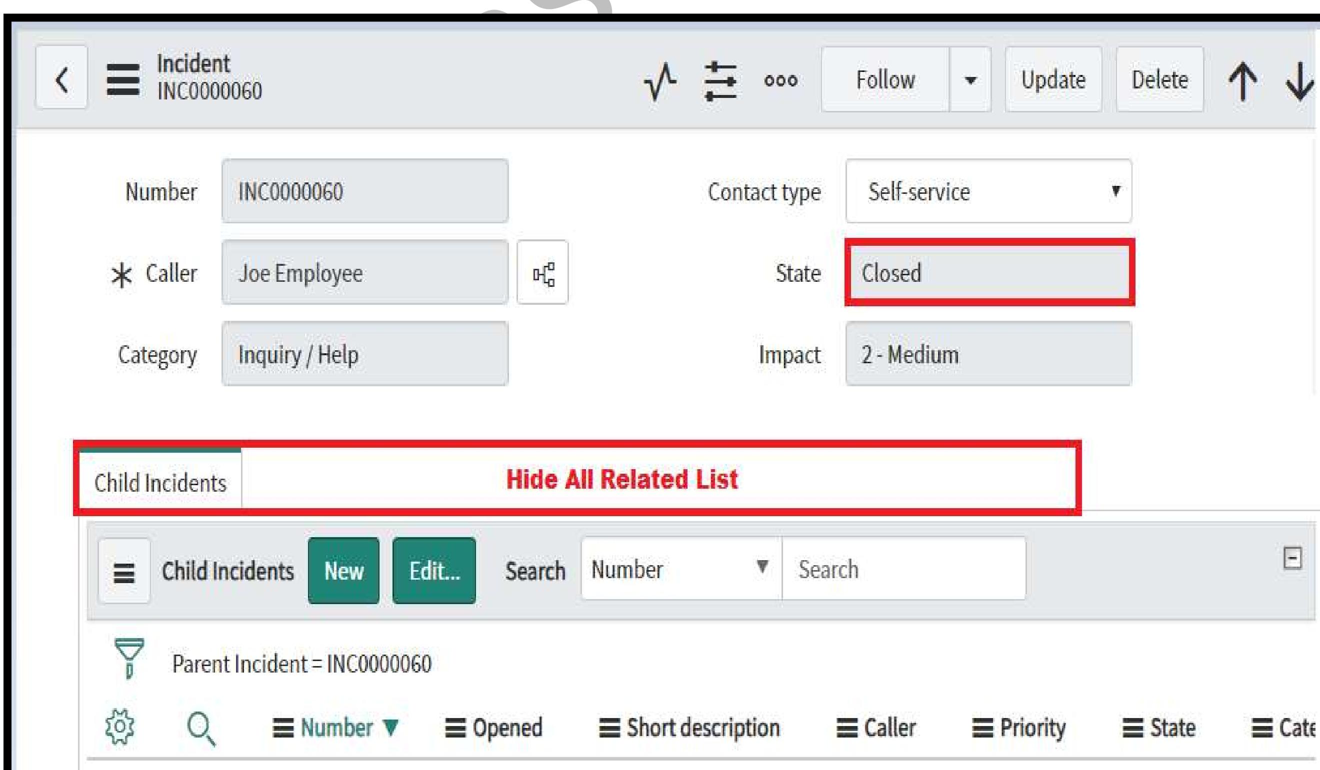


16. Set List Name **Request→Parent** Visible **False**
17. Click on **Submit**





18. Navigate > incident record to see how it is working



Reverse if false:

When we enabled this check box. If the incident state is not **Closed**, the UI policy is reversed.

On load

When we Enabled this check box to perform the actions when the form is **loaded** or when the **condition changes.**

Inherit

Tables that extend the specified **Table** inherit this UI Policy

Use Client scripts in UI policies

We can write **Client Scripts** for **UI policies** that will run on the client side.

We can use the **UI Policy form** to create client scripts that run **Onchange** when the UI policy conditions are met (**Execute if true**) or not met (**Execute if false**). To display these scripting fields in the UI Policy form, in the Script section, **Enabled** the **Run scripts** check box.

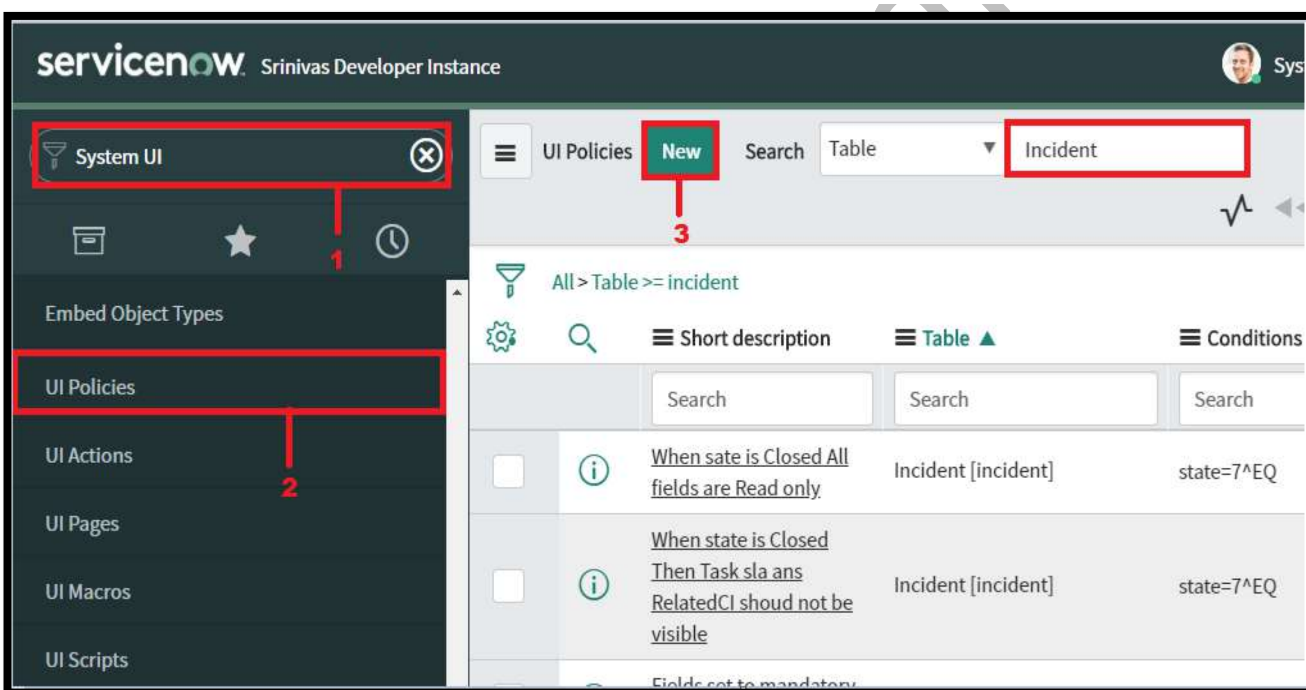
Exercise: 9 About This Requirement

When the incident **State** field value changes to **Closed,** given alert message to customer, if condition is **True**

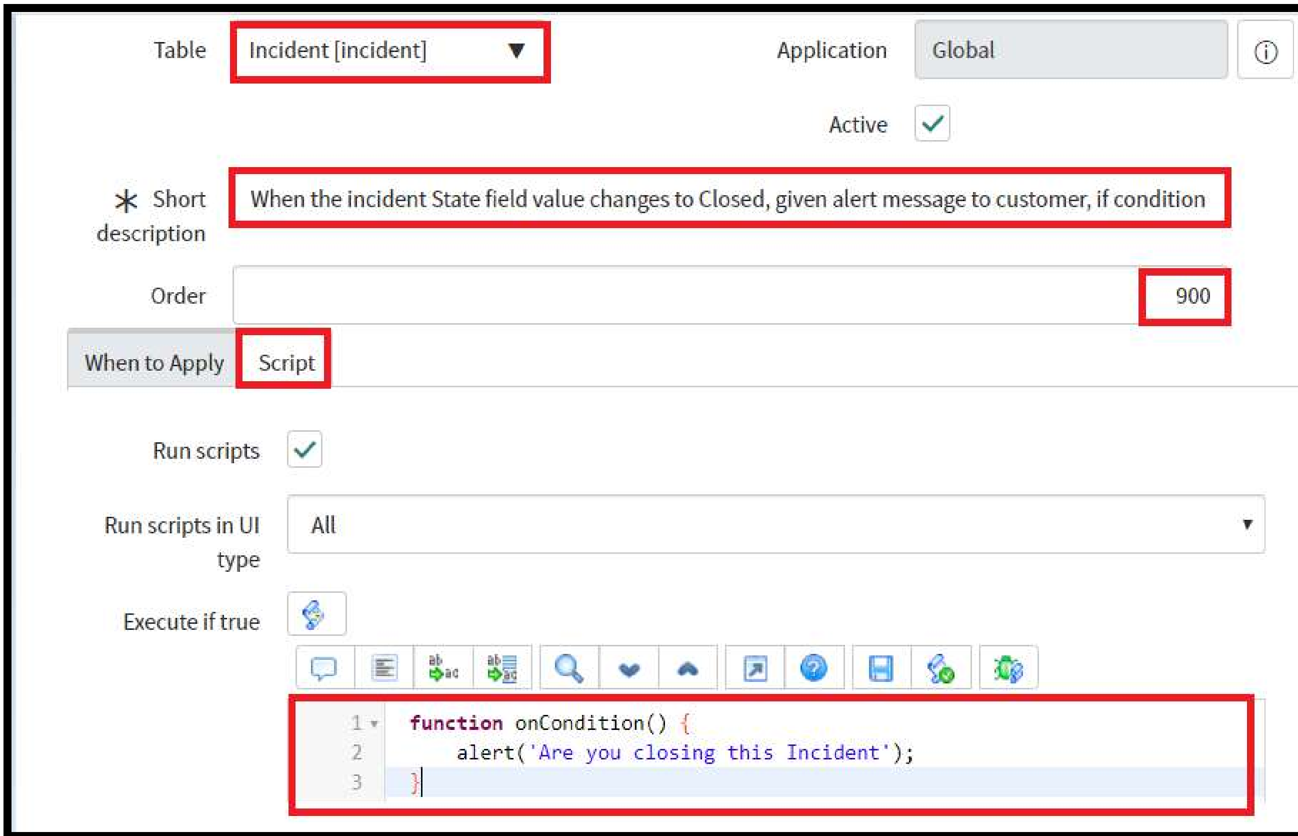
- **Condition** : **[State]** [Is] **[Closed]**
- **Alert Message** : **'**Are you closing this Incident'

Procedure

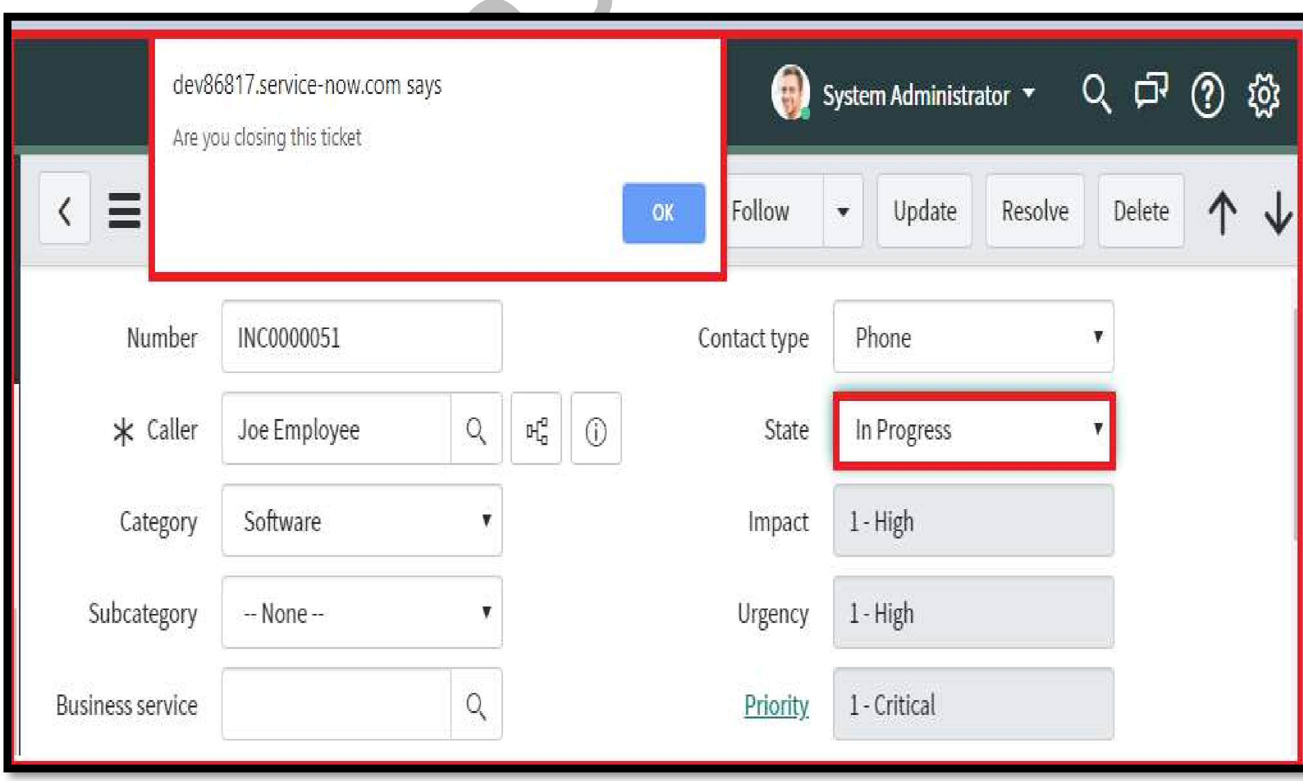
1. Navigate to **System UI** > **UI Policies**.
2. Click **New**.
3. Fill the form with below details



4. **Table:** Incident
5. **Condition: State** is **Closed**
6. **Order** :**900**
7. **Short Description**: When the incident **State** field value changes to **Closed,** given alert message to customer, if condition is **True**
8. Select **Script** Tab
9. **Run scripts: Enabled** Check box
10. **Run Script in UI type:** All
11. Click on **Submit**



12. Navigate > incident record to see how it is working



Chapter – 10

Data Policies

What is Data Policy

Data policy are similar to UI Policy used to enforce data consistency by setting **Mandatory** and **Read-only** states for fields.

Data policies are working as UI policies, but UI policies only apply to data entered on a form through with in the browser. Data policy rules can apply to all data entered into the system, including data brought in through import sets or web services and data entered through the mobile UI enter data through java script.

Example, suppose that we are configured a web service or rest API that allows users from outside the platform to update incident record on the Service Now instance. but these incidents are not updated through the instance UI, they are not subject to the UI policies on the problem form. To ensure that the **Resolution code** and **Resolution notes** field is completed before an incidents are marked **Closed/Resolved**,

1. We can create a data policy that applies to **server-side imports.**
2. Data that does not comply with this data policy displaying an error.
3. We can also apply the **UI policy** on the browser by selecting the **Use as UI Policy on client** check box in the data policy record.
4. By default, data policies are applied to all **Glide Record** operations including those used in Scripted REST APIs, and the REST Table API.

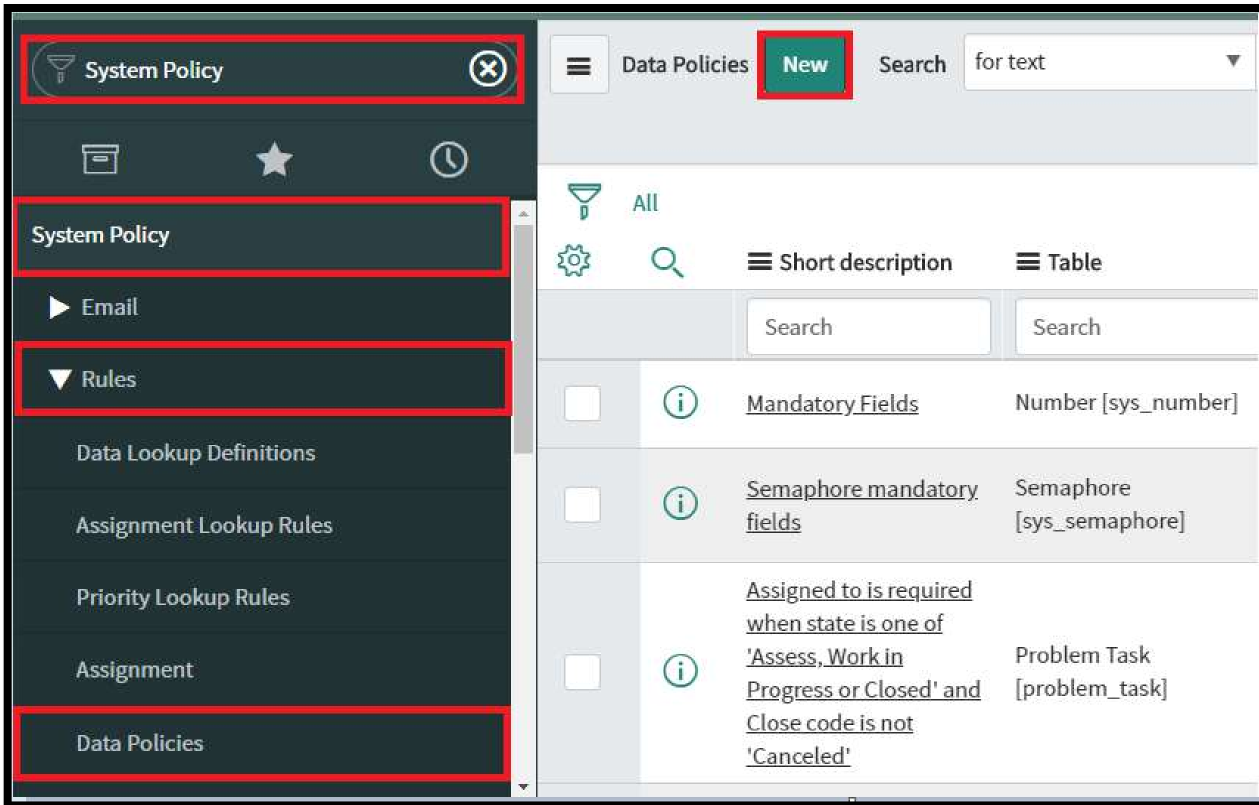
Exercise :1 Create New Data Policy

We can create a new data policy rule to define data rules for a **Incident** table.

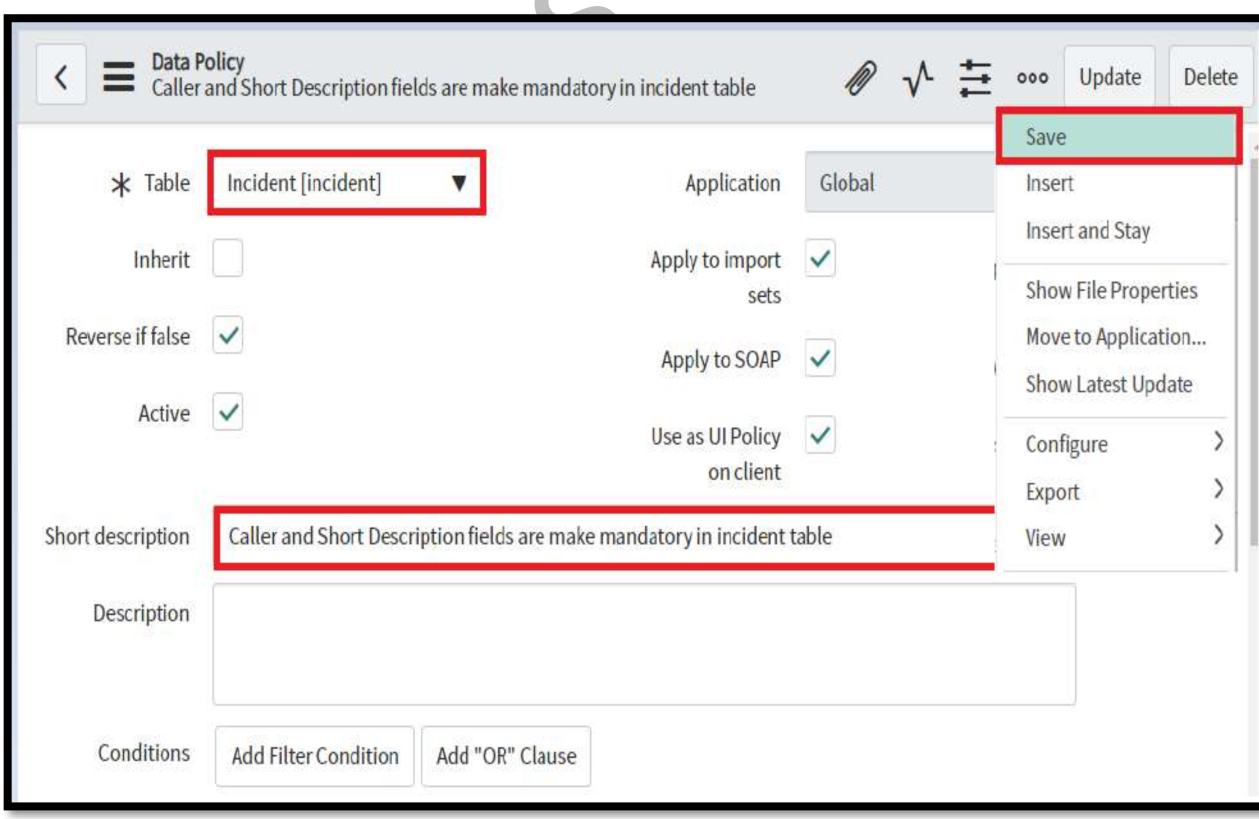
Caller and **Short Description** fields are make mandatory in incident table

Procedure

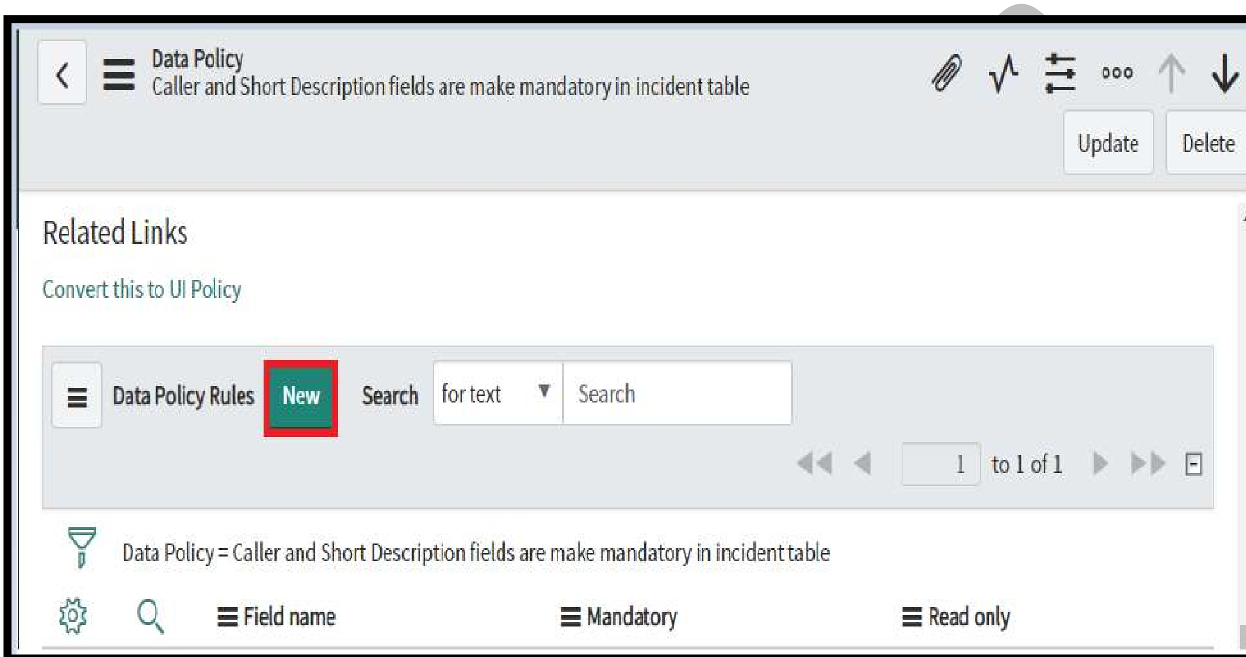
1. Navigate to **System Policy** > **Rules** > **Data Policies**.
2. Click on **New**



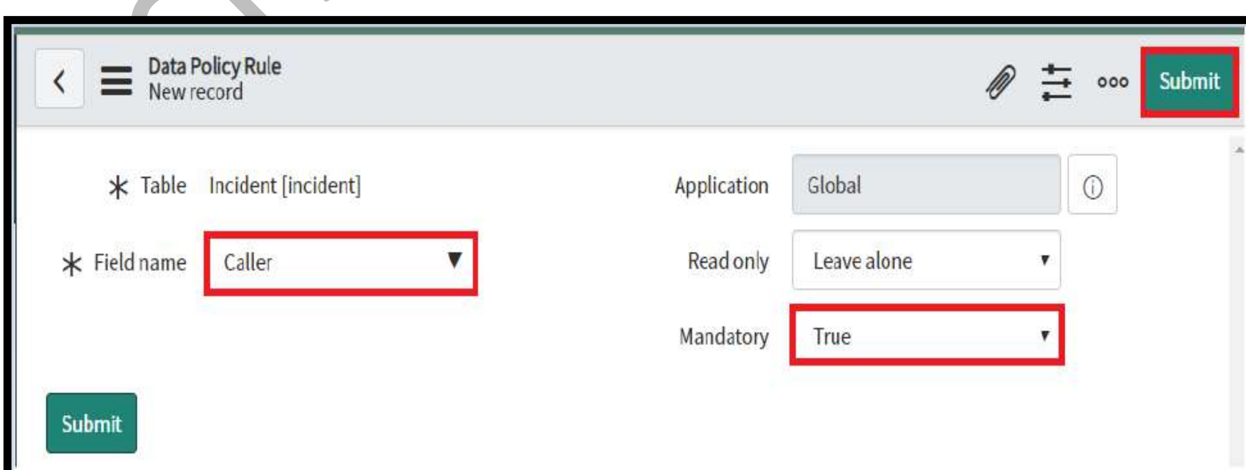
3. Fill **Data Policy** form



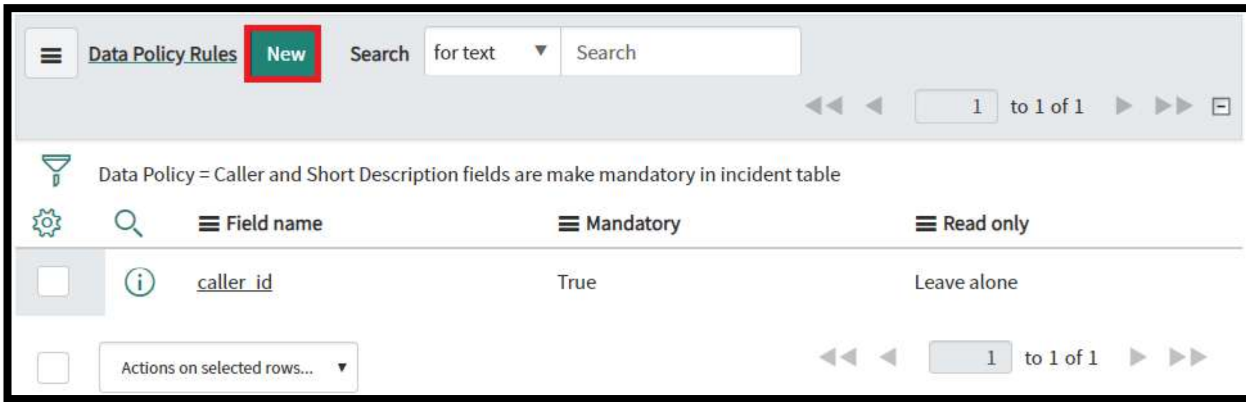
4. **Table:** Incident
5. **Reverse if false:** Auto Filled
6. **Active:** Auto filled
7. **Apply import sets:** Auto filled
8. **Apply to REST/SOAP:** Auto filled
9. **Use as UI Policy on Client:** Auto filled
10. **Short Description: Caller** and **Short Description** fields are make mandatory in incident table
11. **Right click** on form Header and **Save**



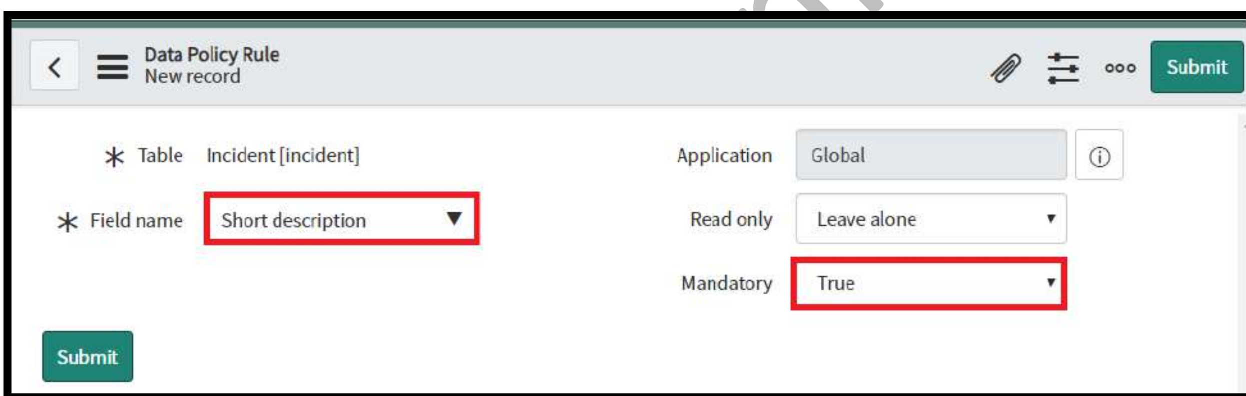
12. Come down on same form
13. Create New **Data Policy Rule** click on **New**



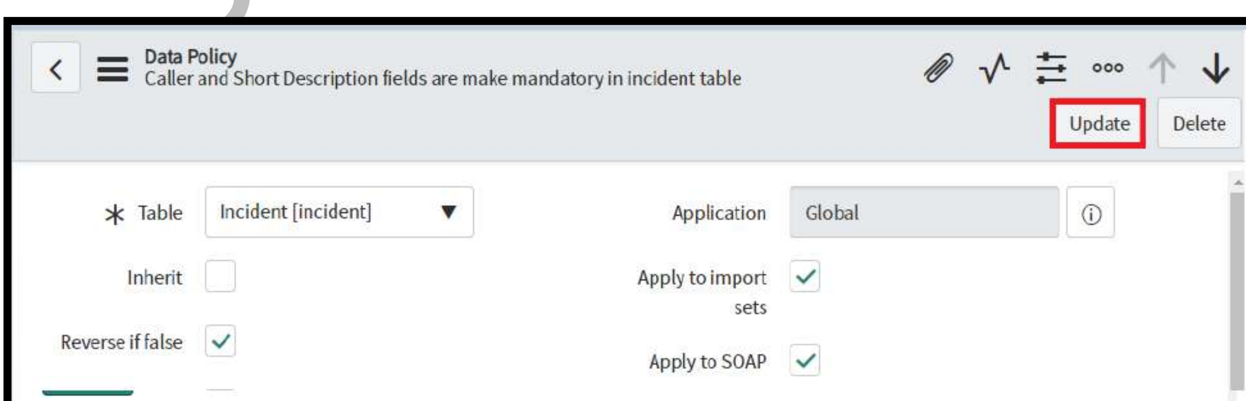
14. Field Name: Caller
15. Mandatory: True
16. Click on **Submit**



17. Create New **Data Policy Rule** click on **New**



18. Field name: Short Description
19. Mandatory: True
20. Click on **Submit**
21. Click on **Update**



Exercise 2: Converting UI policy to data policy

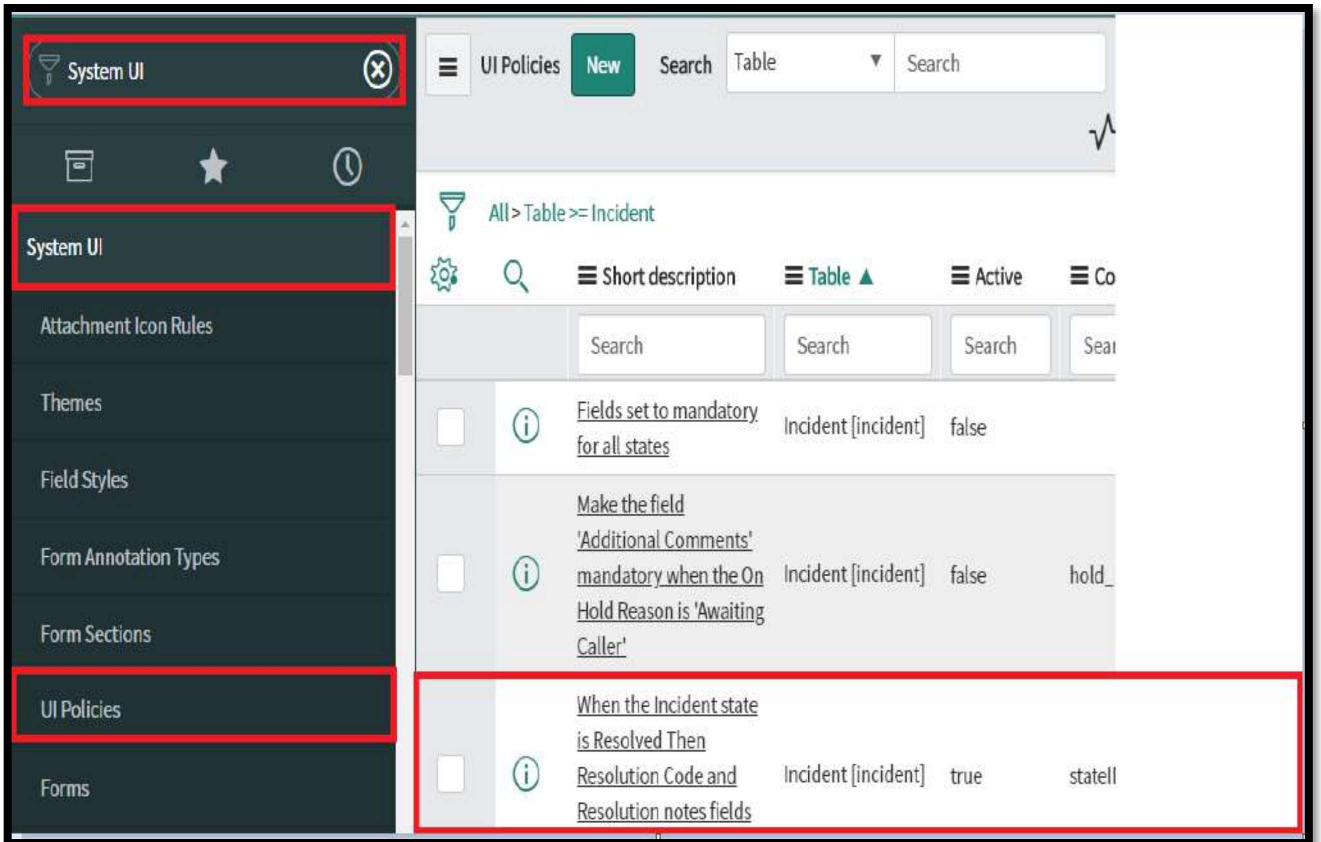
We are going to convert the **UI policy** to **Data policy** to apply the same condition to data policy for server side

Requirement: When State is **Resolve** Then **Resolution code** and **Resolution note** are mandatory

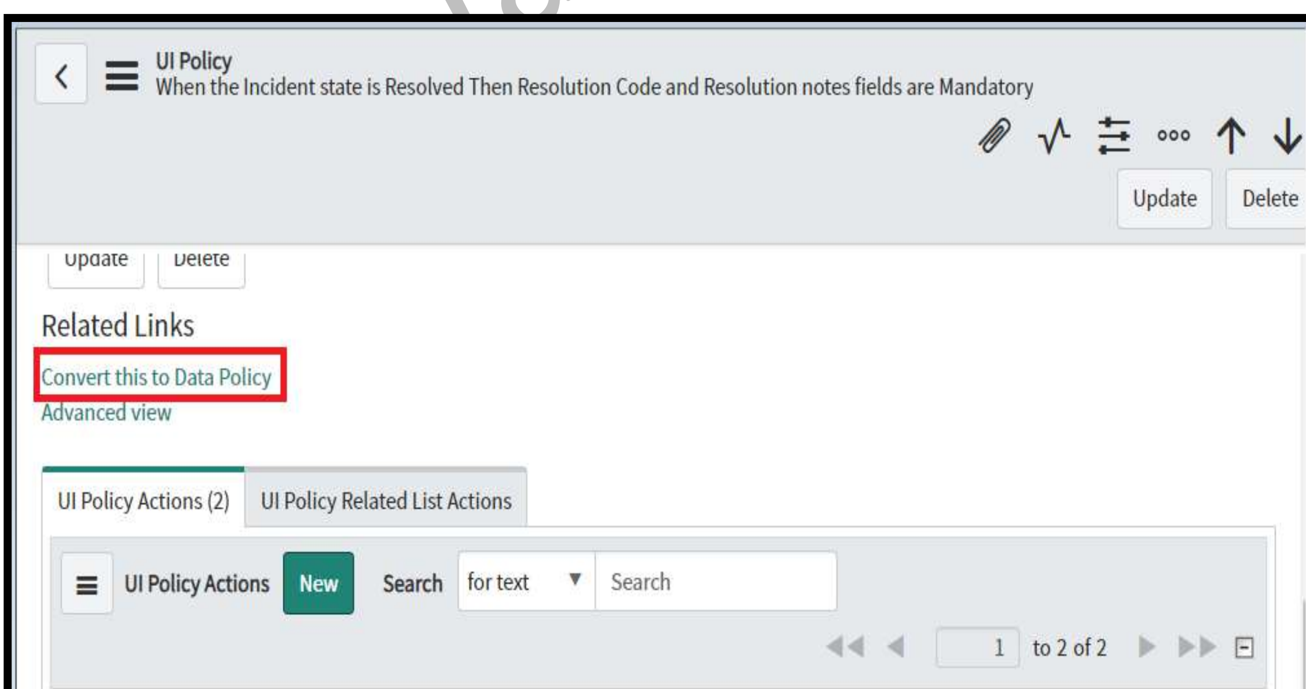
1. We can also apply a UI policy condition and action to **Import Sets**
2. Data imported by **REST/SOAP** web services when we convert it to a data policy.
3. When we create a record from **Glide Record API**
4. When we can convert a **UI policy to a data policy** then automatically deactivates the UI policy record. activates data policy record
5. To retain the policy in the UI, ensure that the **Use as UI Policy on client** check box is selected on the data policy record.
6. While converting UI policy to be eligible for conversion to a data policy, the following conditions should be met on the **UI Policy form**.

Procedure

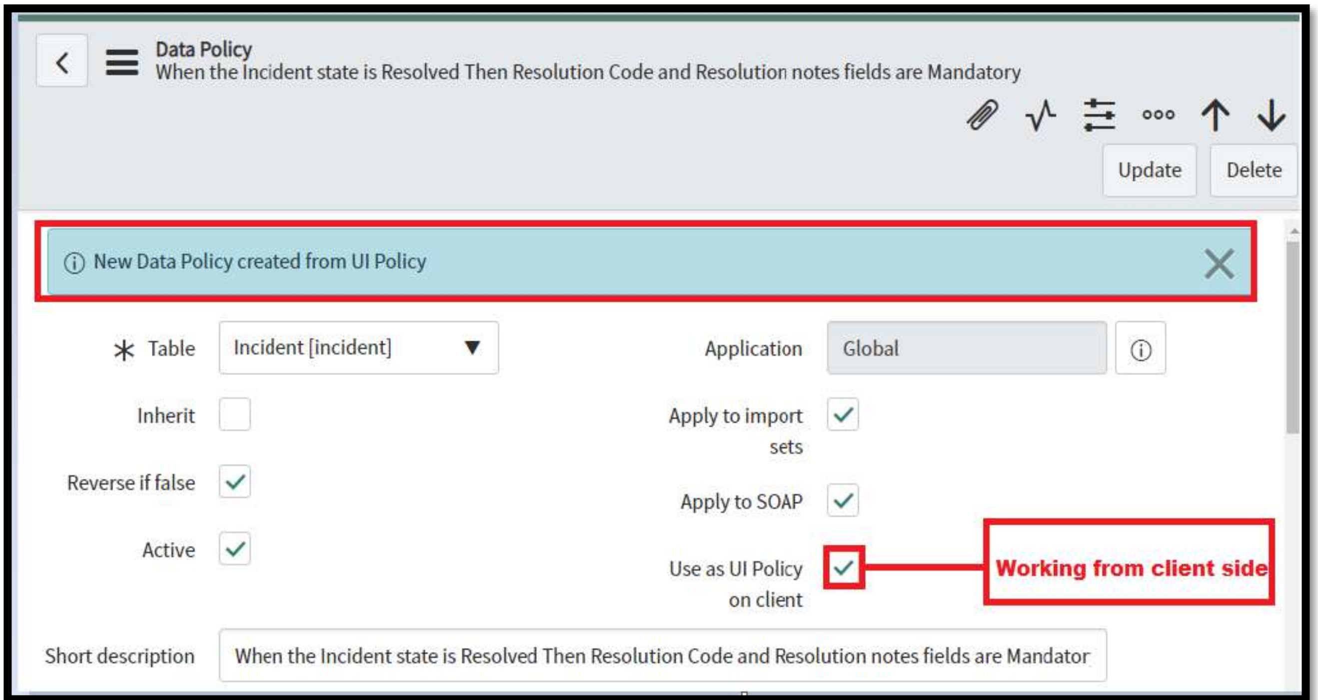
1. Navigate to **System UI** > **UI Policies**
2. Open an existing **UI policy** record like below screen shot



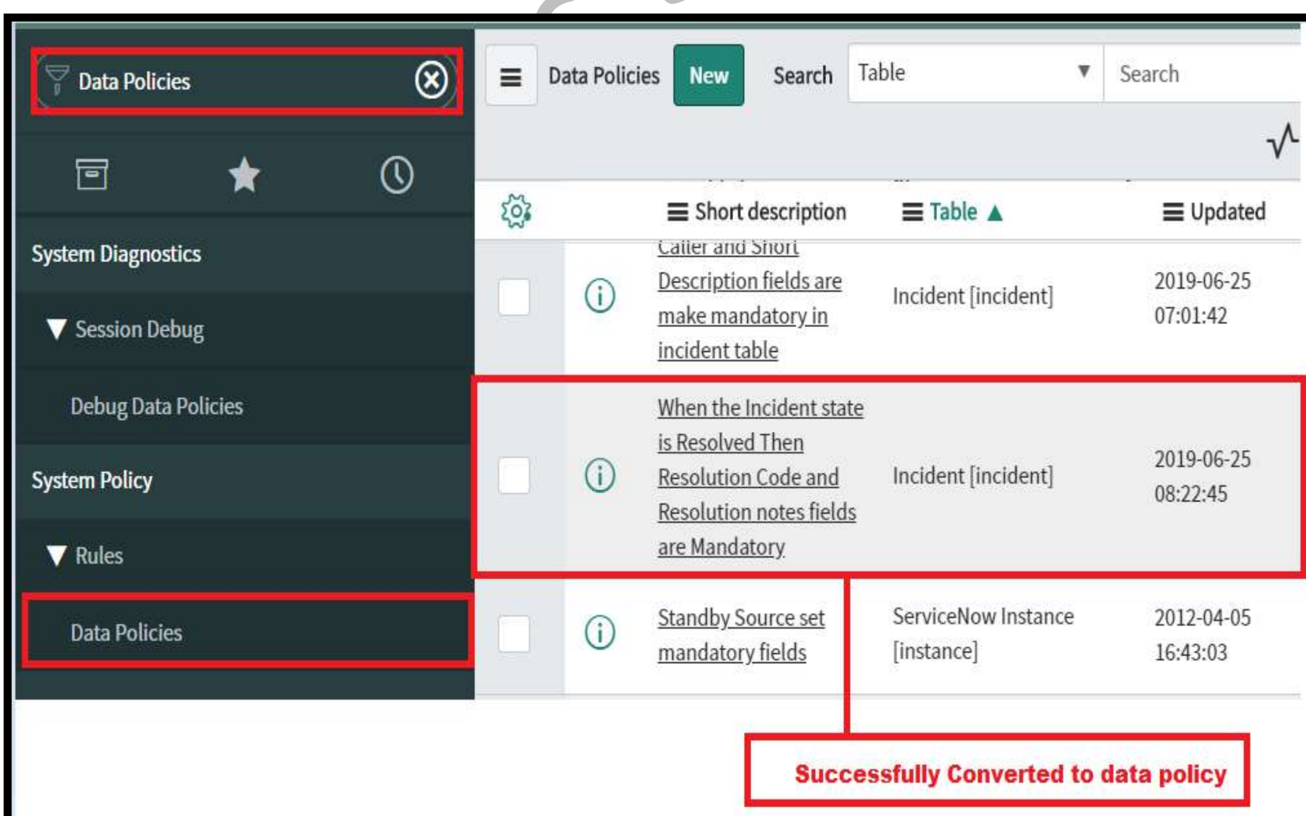
3. Go to under **Related Links**,
4. Click on **Convert this to Data Policy**



5. We will get below screen shot after converted



6. Click on **Update**



7. Refer below screen shot successfully converted to **Data Policy Rule**

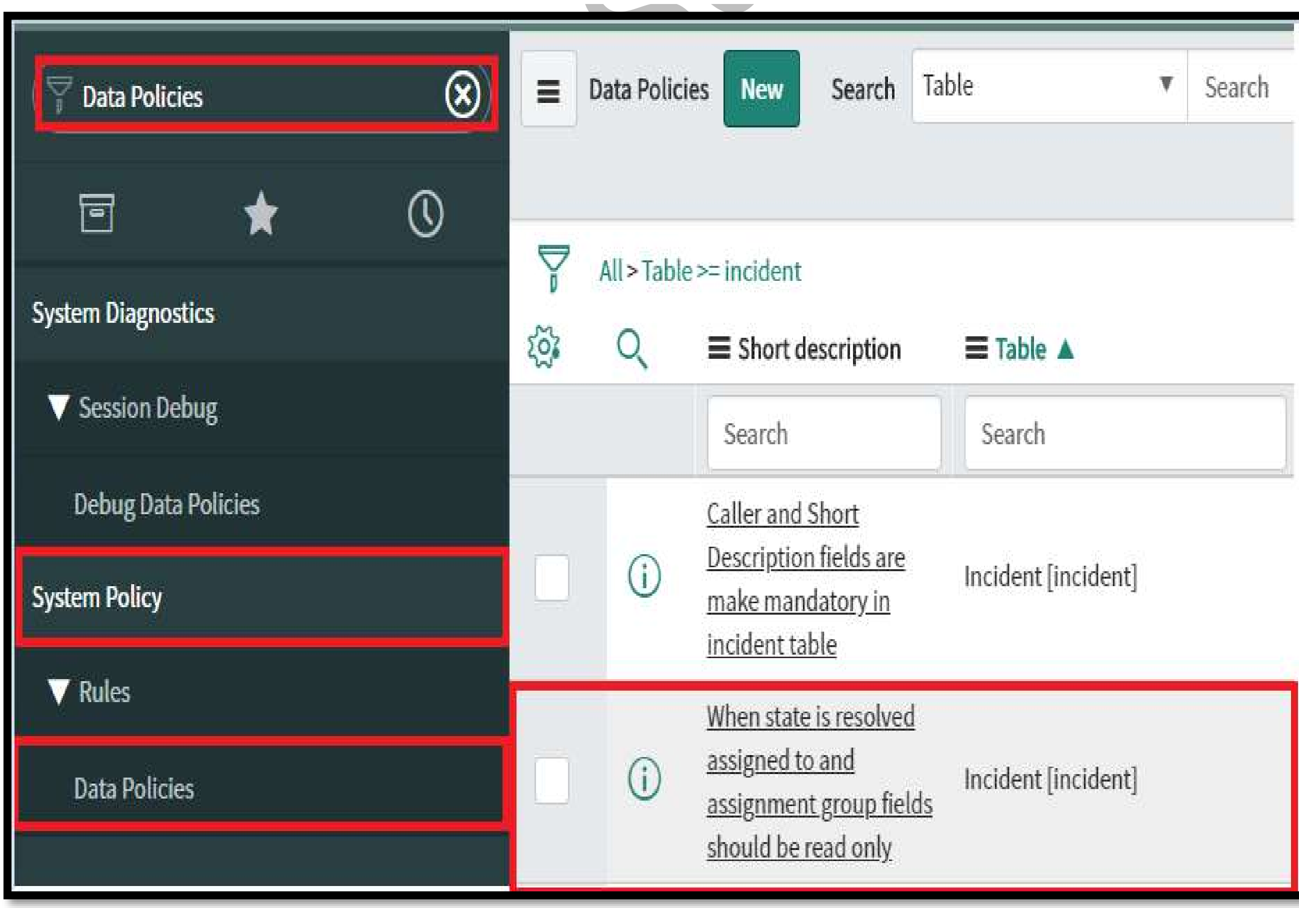
Exercise: 3 Convert Data policy to UI policy

We can convert a data policy to a UI policy is useful if a data policy already exists, but only needs to apply to records created or updated with in browser and client side. While convert to data policy this is not deactivates - the new UI policy will have applied only at the system UI browser side and not working to import sets or data imported from REST/SOAP web services

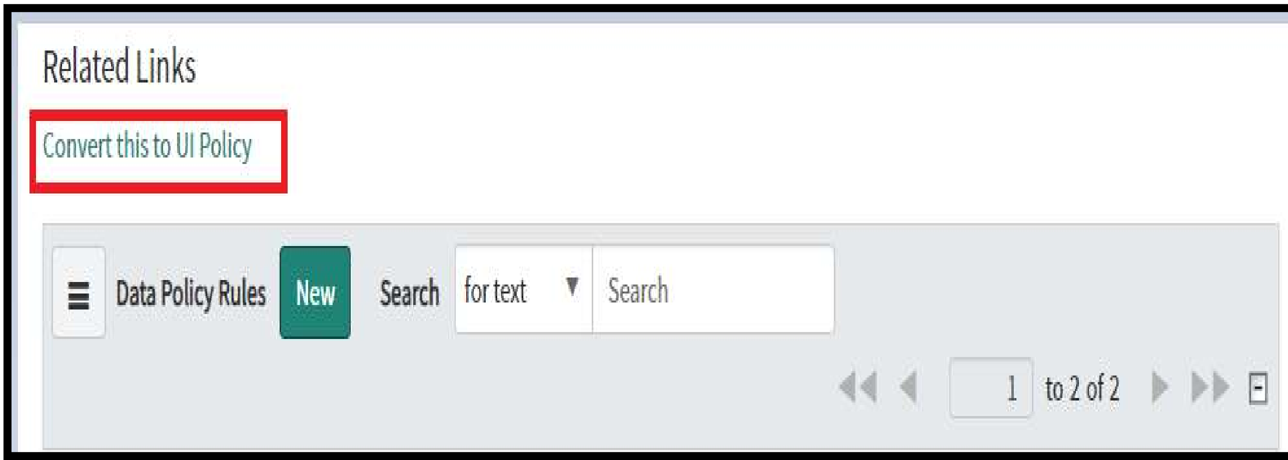
1. This is an alternative to converting from a **Data policy to a UI policy** is to select the Use as UI Policy on client checkbox on the data policy record.
2. This field extends the data policy to the UI. The main difference between converting and using the Use as UI Policy on client checkbox is that converting provides the Visible field on the UI policy record.

Procedure

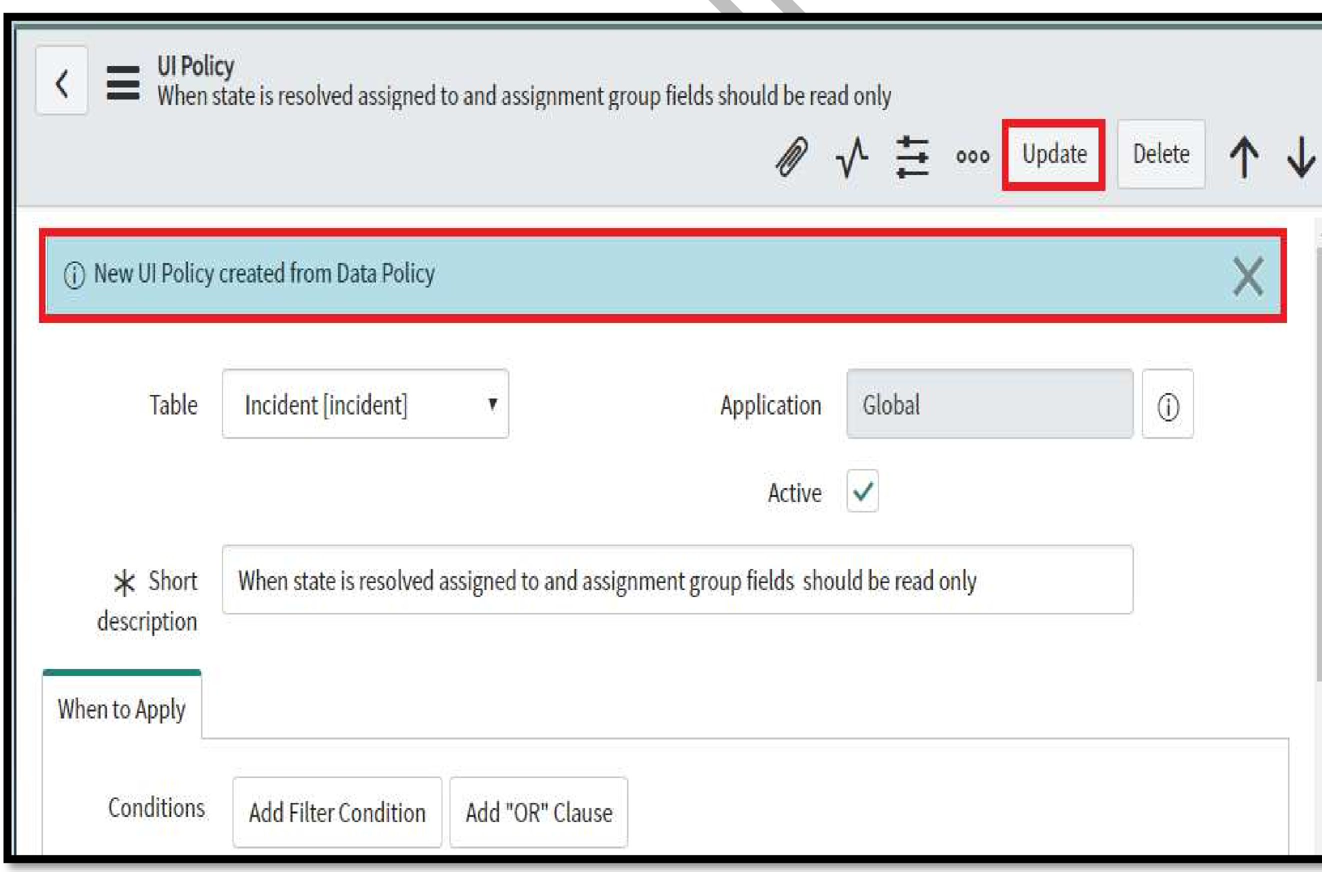
1. Navigate to **System Policy** > **Rules** > **Data Policies**.
2. Open existing **Data Policy** record



3. Go to Under **Related Links**,
4. Click on **Convert this to UI Policy**.



5. Refer below screen shot successfully converted to **UI Policy**
6. Click on **Update**



Chapter – 11

Metrics

What is Metrics?

A metrics can be calculating and evaluates the effectiveness of IT service management processes in Business.

Ex: a metric will measure the effectiveness of the incident, resolution process by calculating how long it takes to resolve an incident and close the incident

Highlights of Metrics

1. Sometimes a metric can be easily gain from the data.
2. It is used to find the number of incidents or problems that were resolved or closed today,
3. We can generate a report will simply count the number of incidents in the incident table with a Created date of last 3 months.
4. Often, however, metrics need to be gathered as **data is updated.**
5. Representing how long an incident was assigned to a certain group requires collecting information about assignment changes and calculating the duration of each assignment.
6. **Ex:** The **"Assignment Group Duration"** metric can calculating the duration of time an incident is assigned to a one group to another group.
7. The metric is defined by creating a metric definition of type **"Field value duration"** and selecting the "Assigned to" field from the Incident table.

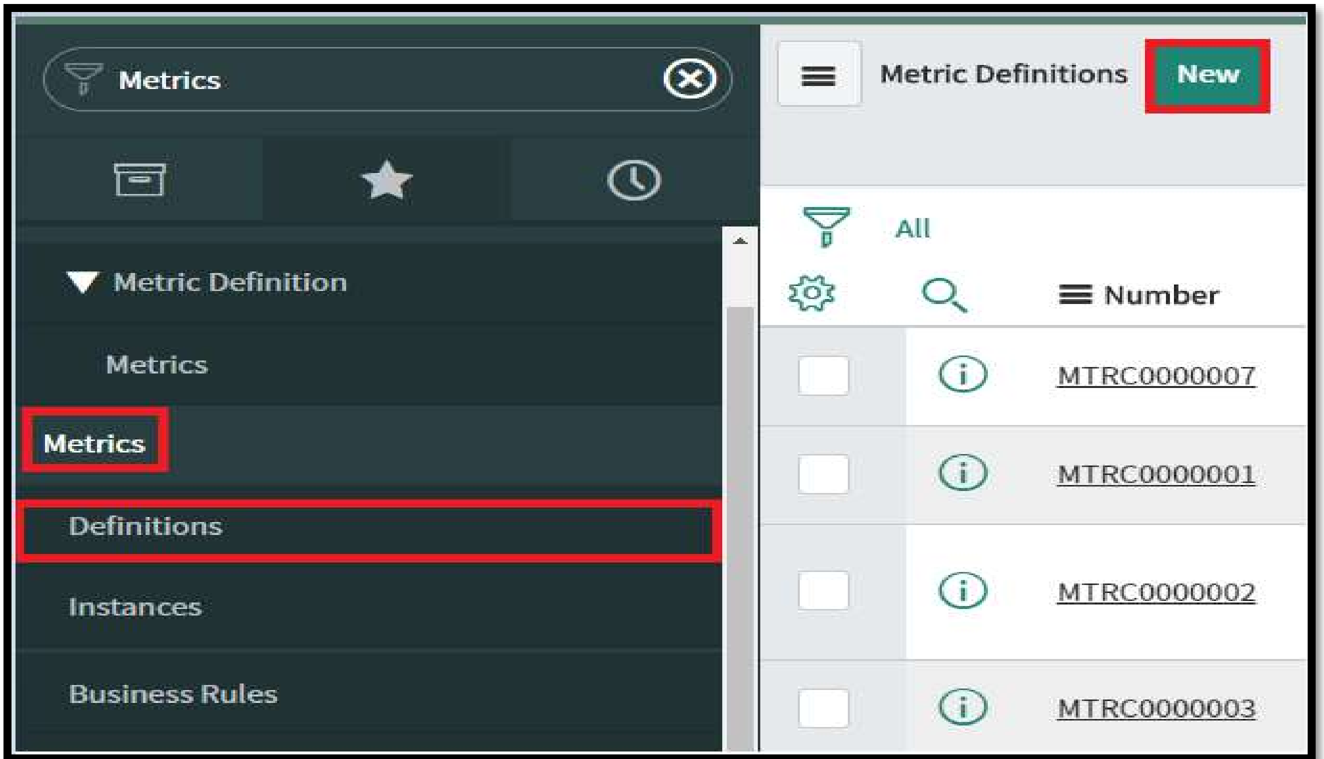
Exercise :1 Create a metric definition

Create a metric definition for **Incident** table

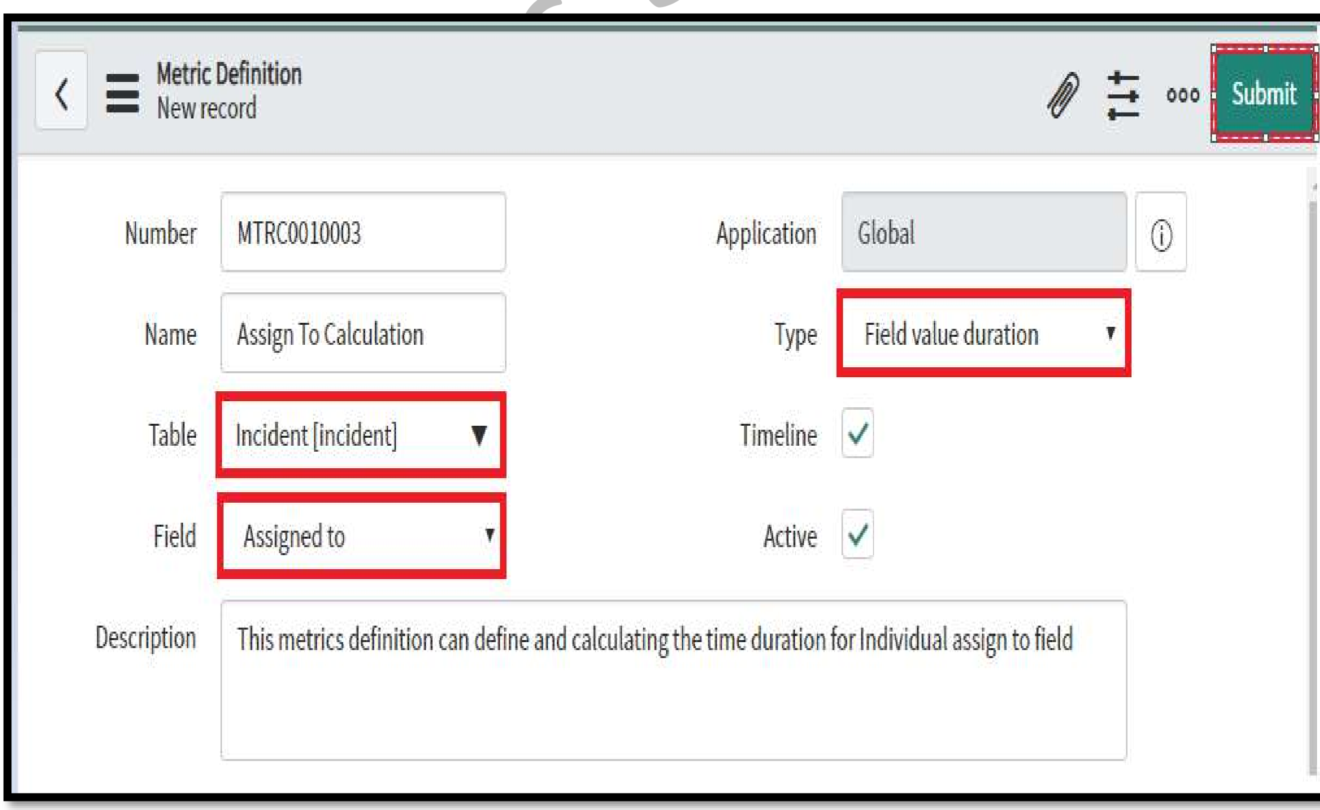
This metrics can be calculating the time duration between Individual **assign to** field

Procedure

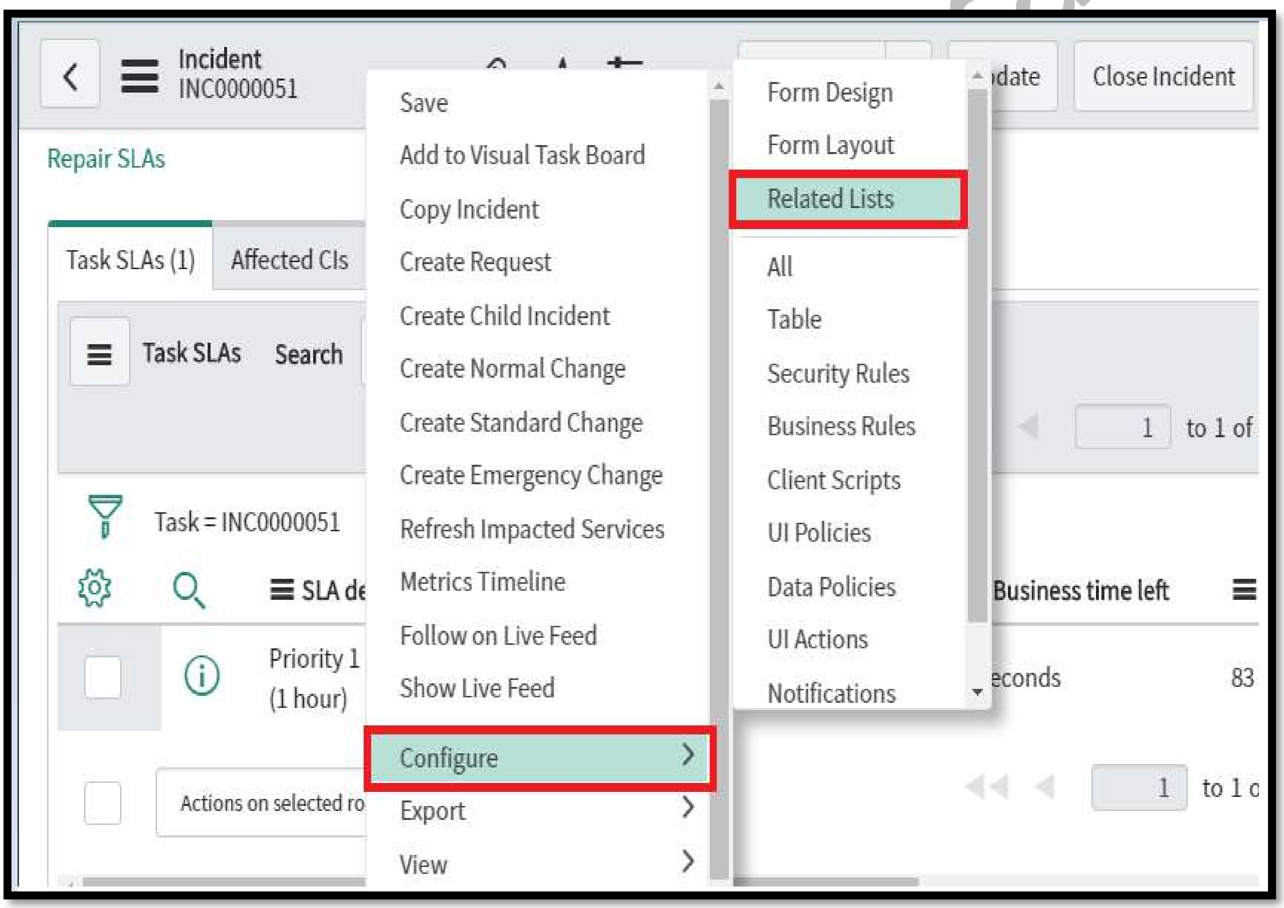
1. Navigate to **Metrics** > **Definitions**
2. Click on **New.**



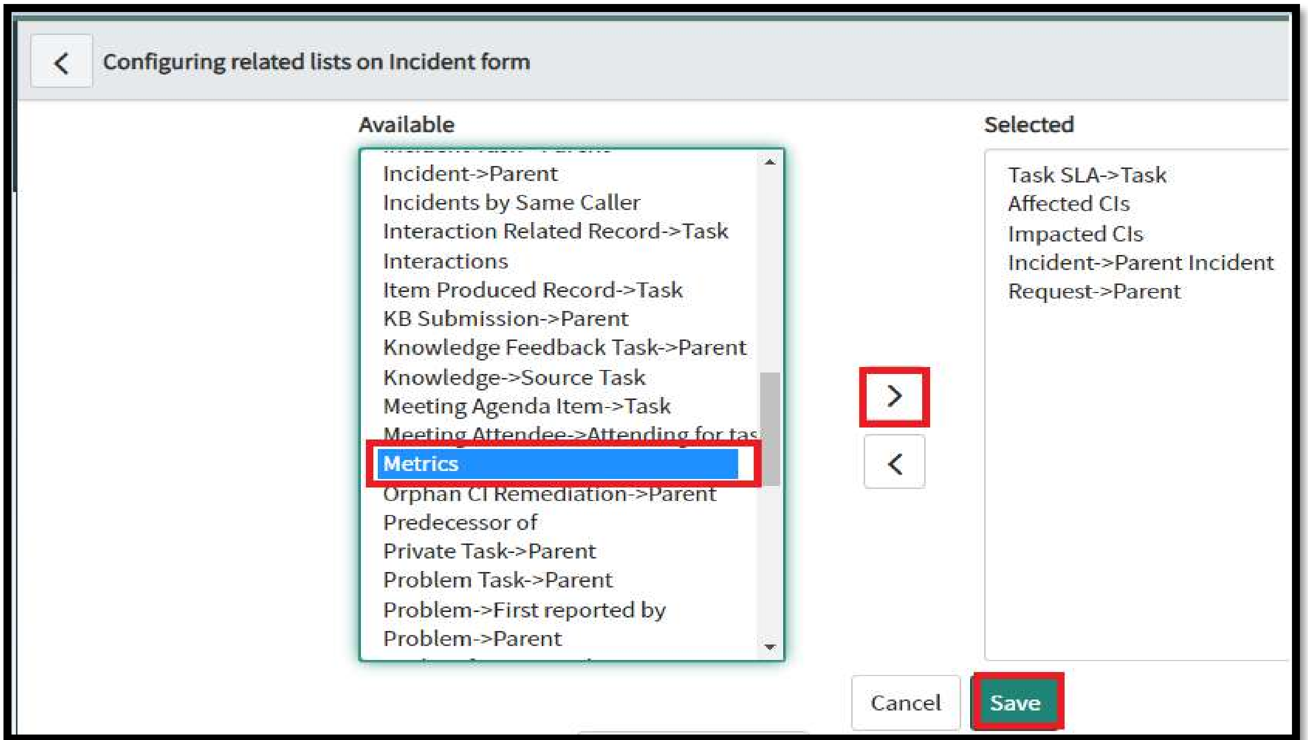
3. Fill metrics definition **Form**



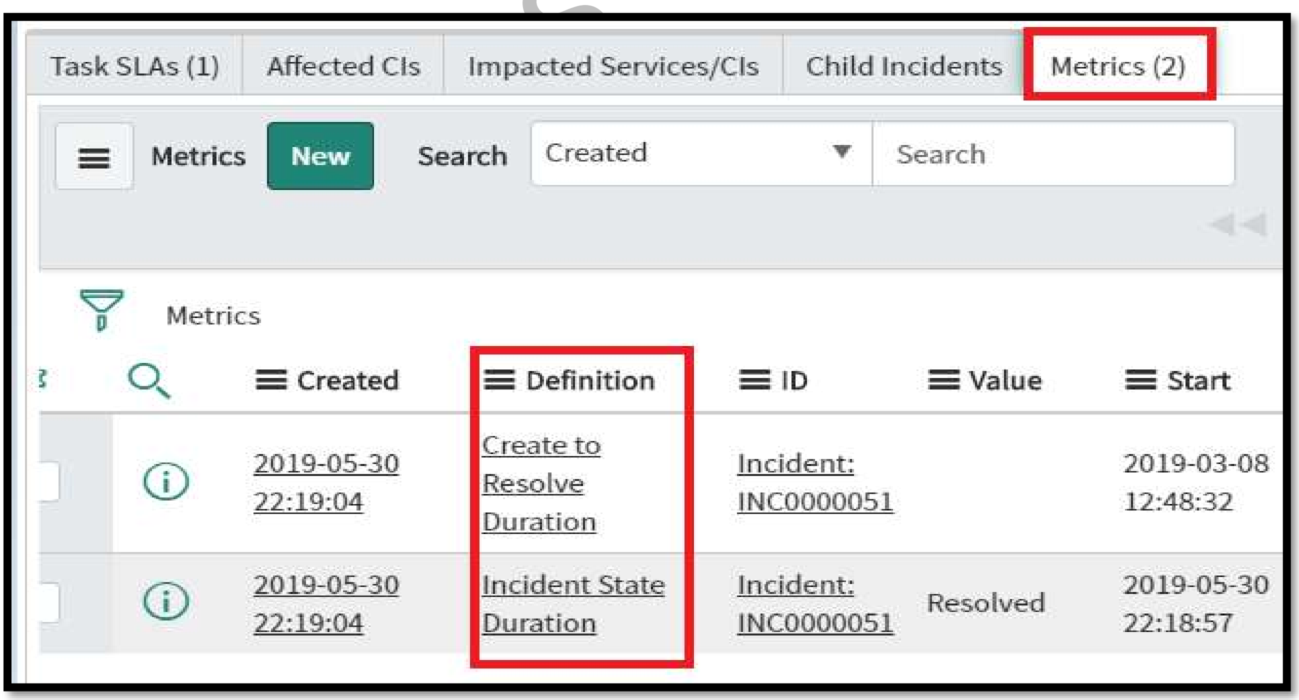
4. **Number:** Auto Filled
5. **Name:** Assign to Calculation
6. **Table:** Incident
7. **Field:** Assign to
8. **Type:** Field value duration
9. **Description:** This metrics can be calculating the time duration between Individual **assign to** persons
10. **Timeline:** check **true**
11. Click on **Submit**
12. Open one existing **incident record** add **Metrics** related list



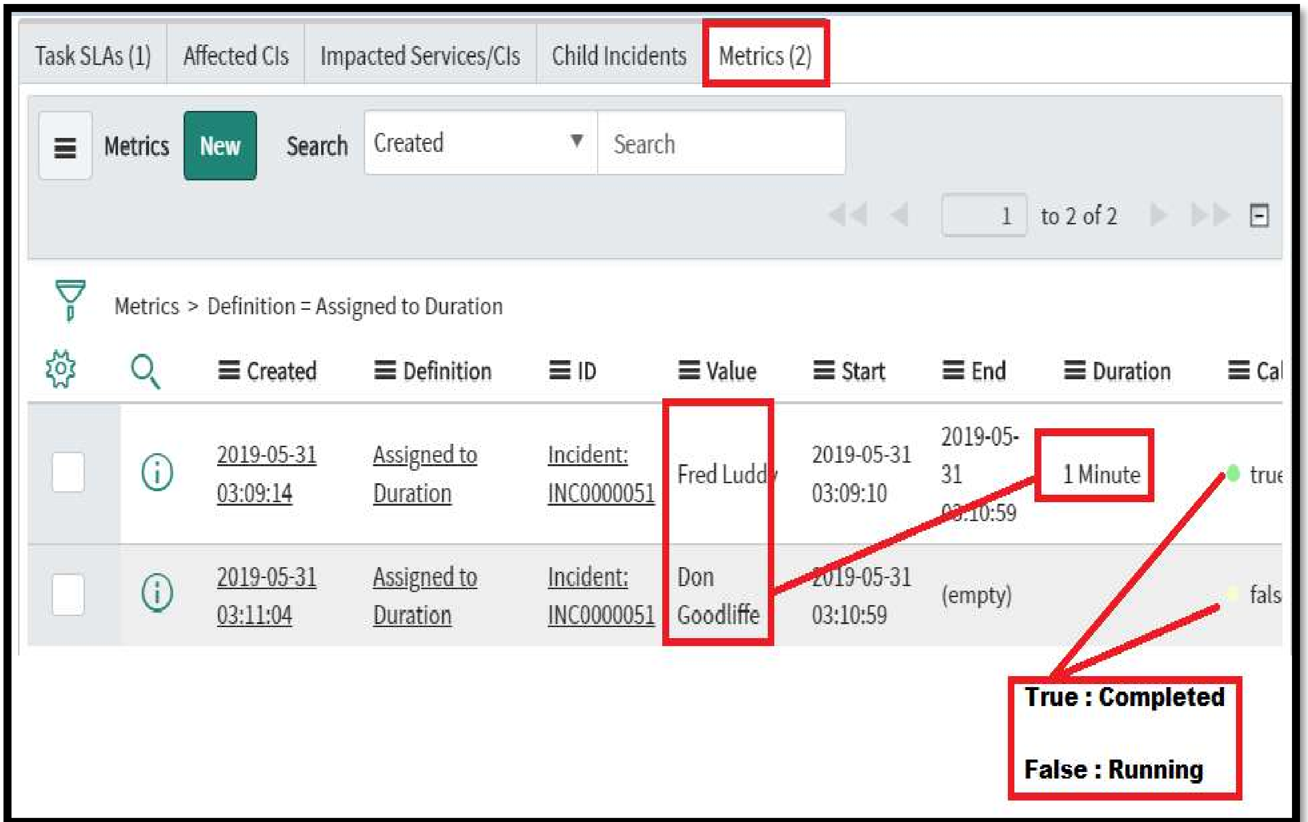
13. Right click on **Form Header > Configure > Related List**



14. Select **Metrics** related List and Click on **> Add**
15. Click on **Save**

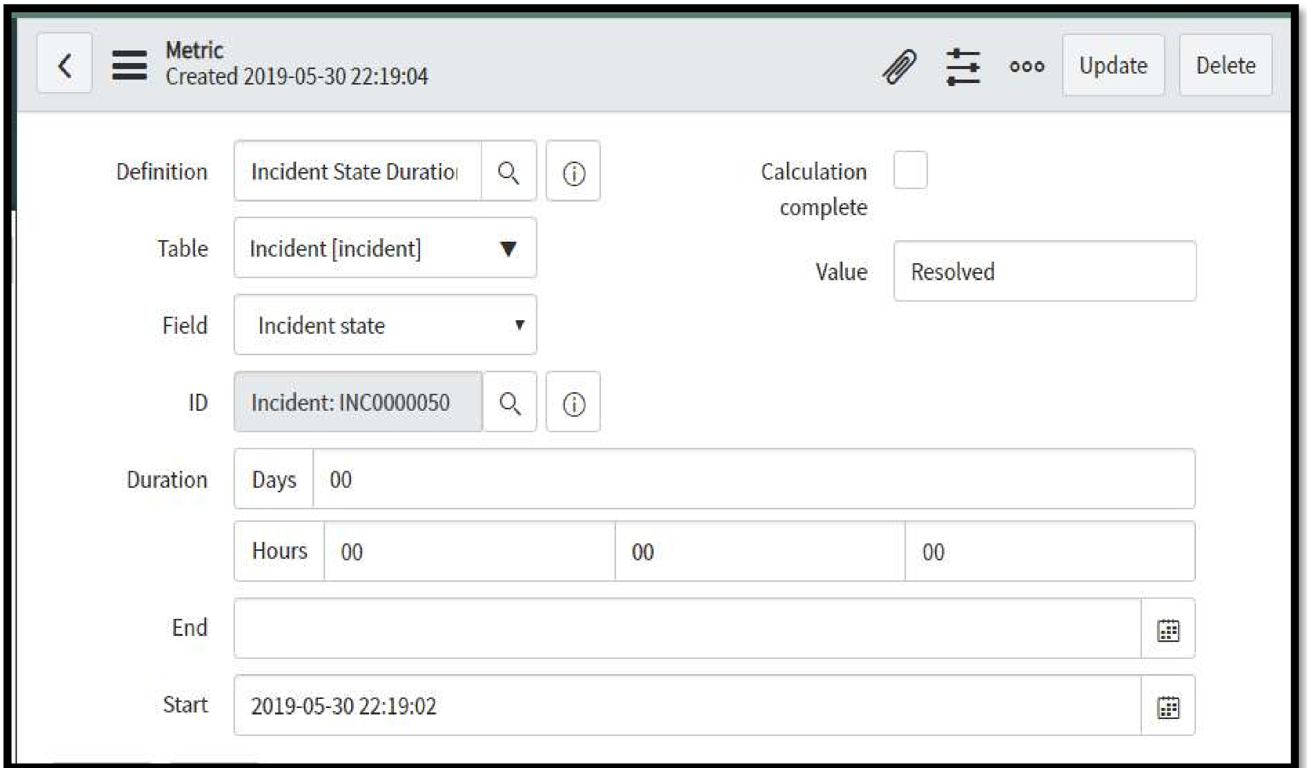


16. Successfully added **Metrics** Related List to **Incident** form
17. Change **assigned to** field value and check **Metrics** calculation



In the base system, metrics are configured to work on the **task** table only. To apply metrics to **cmdb_ci** tables, duplicate the metric events business rule that currently runs on the **task** table for the **cmdb_ci** table. Without the events created, no metric processing

A metric instance is a record in the **metric_instance** table. A record contains details about one instance of a metric.


Metric
Created 2019-05-30 22:19:04
Update
Delete
Definition
Incident State Duratio
Calculation complete
Table
Incident [incident]
Value
Resolved
Field
Incident state
ID
Incident: INC0000050
Duration
Days
00
Hours
00
00
00
End
Start
2019-05-30 22:19:02

Chapter – 12

Related Lists

What is Related List?

The **Related Lists** are showing on particular forms and show records in tables that have relationships to the **current record.**

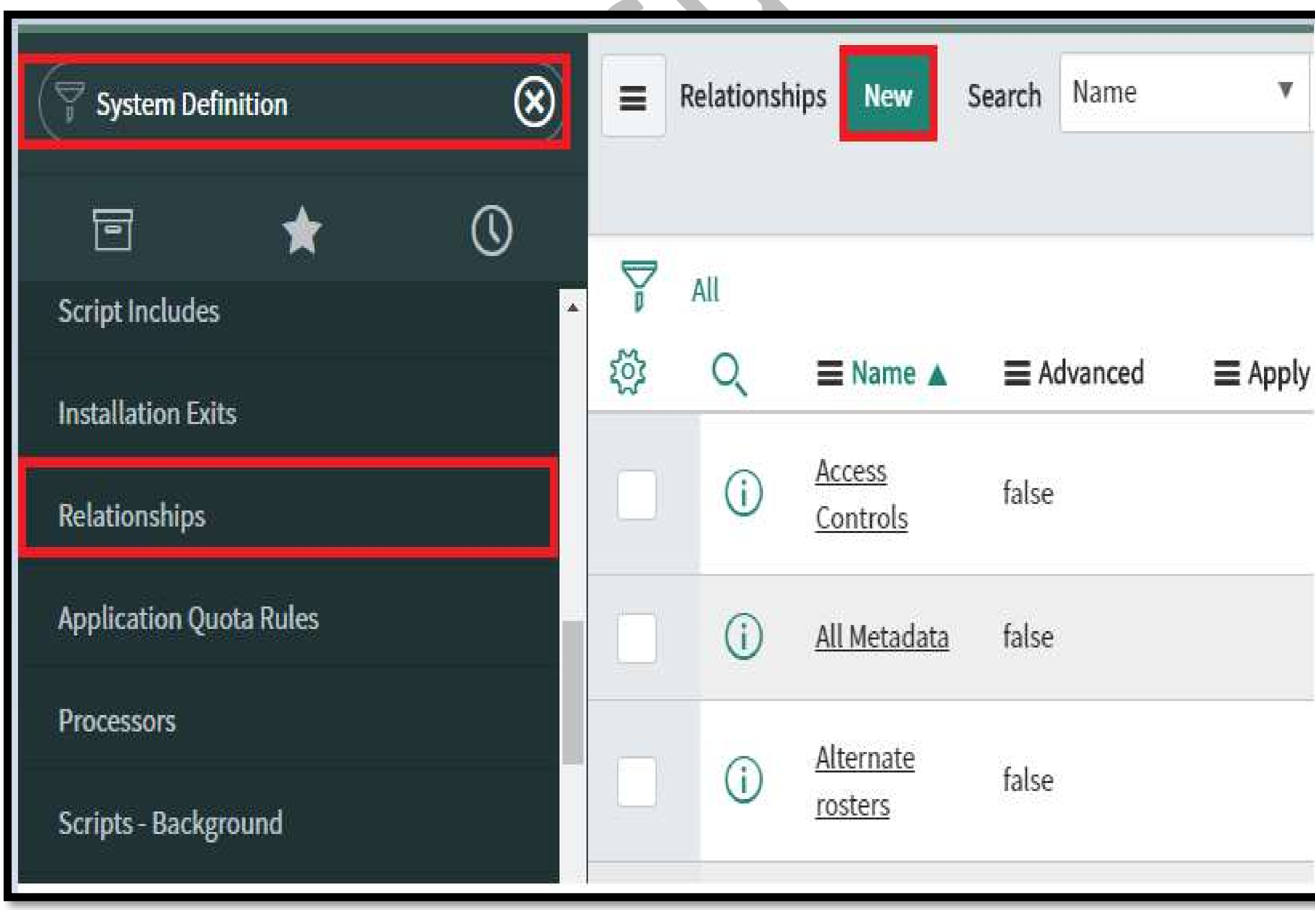
1. Users can view and modify the information in related lists based on current form
2. Only **Administrators** can configure related lists to appear on forms and in hierarchical lists by **configuring a form.**
3. There is no size limitation for related list

Exercise: 1 Configure Related List

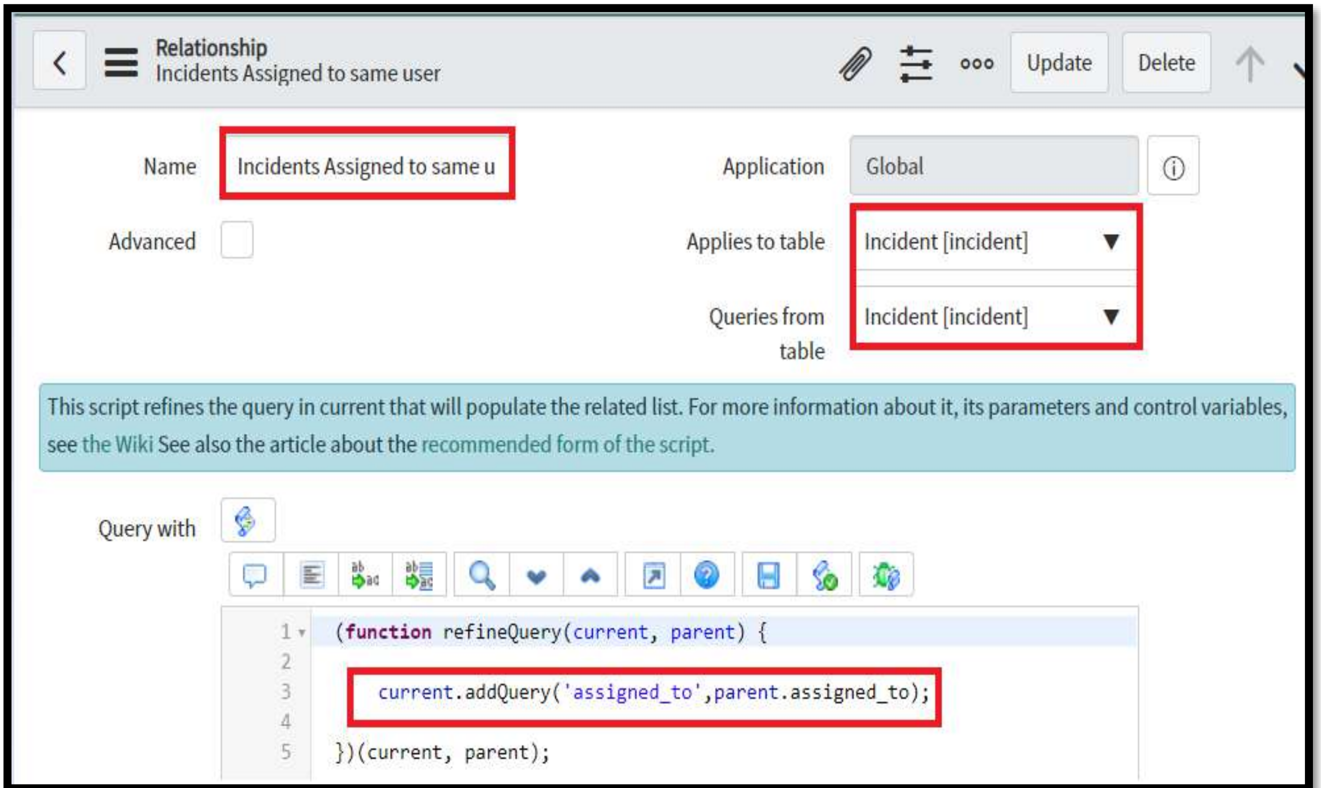
Create a new **Relationship** that will be display all related records based on current **Assign to** field

Procedure

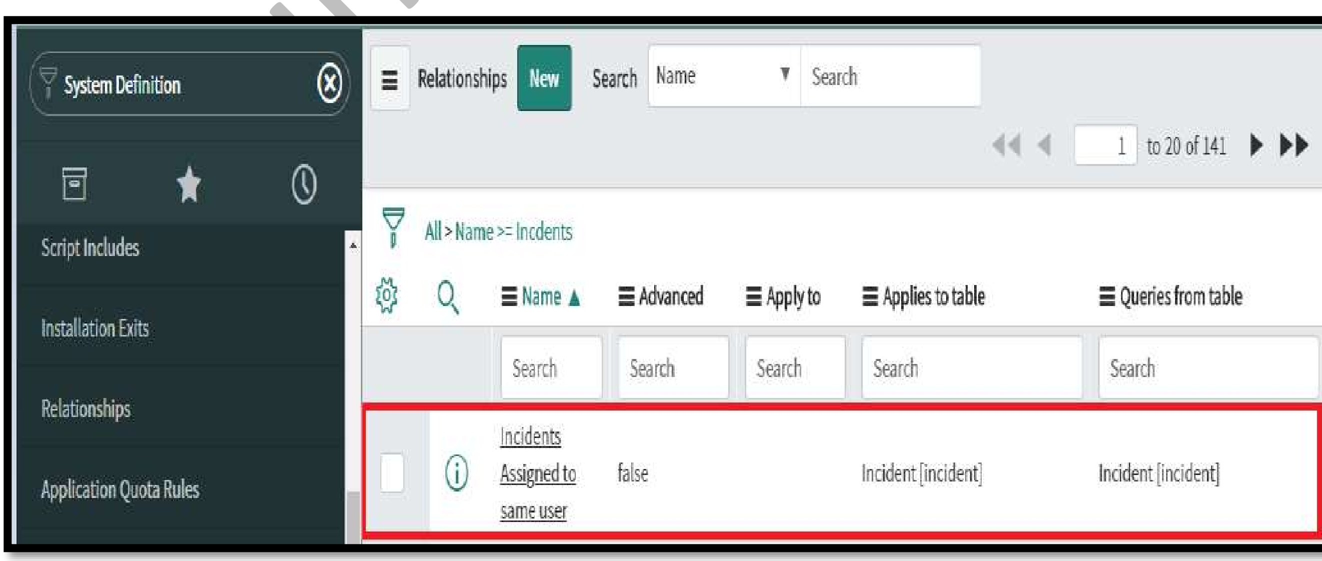
1. Navigate to **System Definition > Relationships**
2. Click on **New**



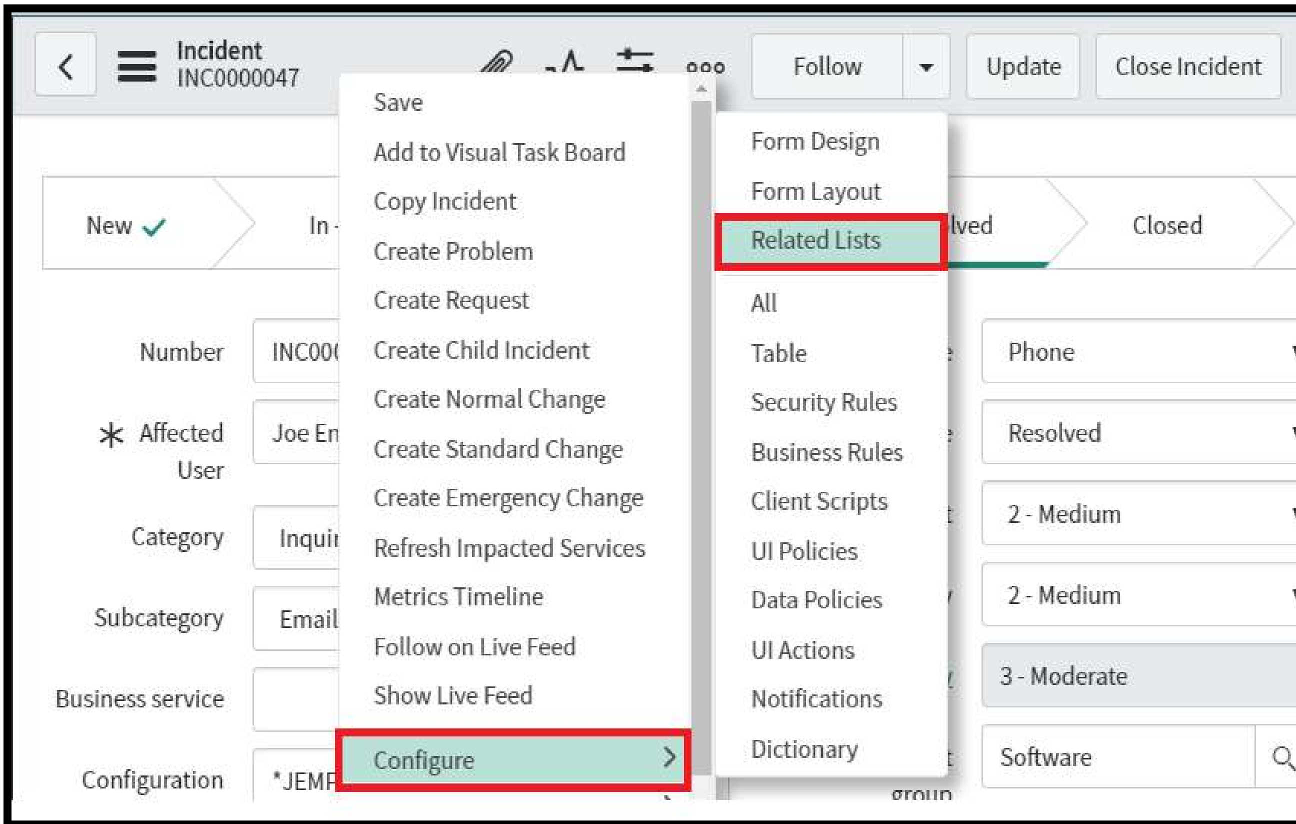
3. Fill **Relationship** form



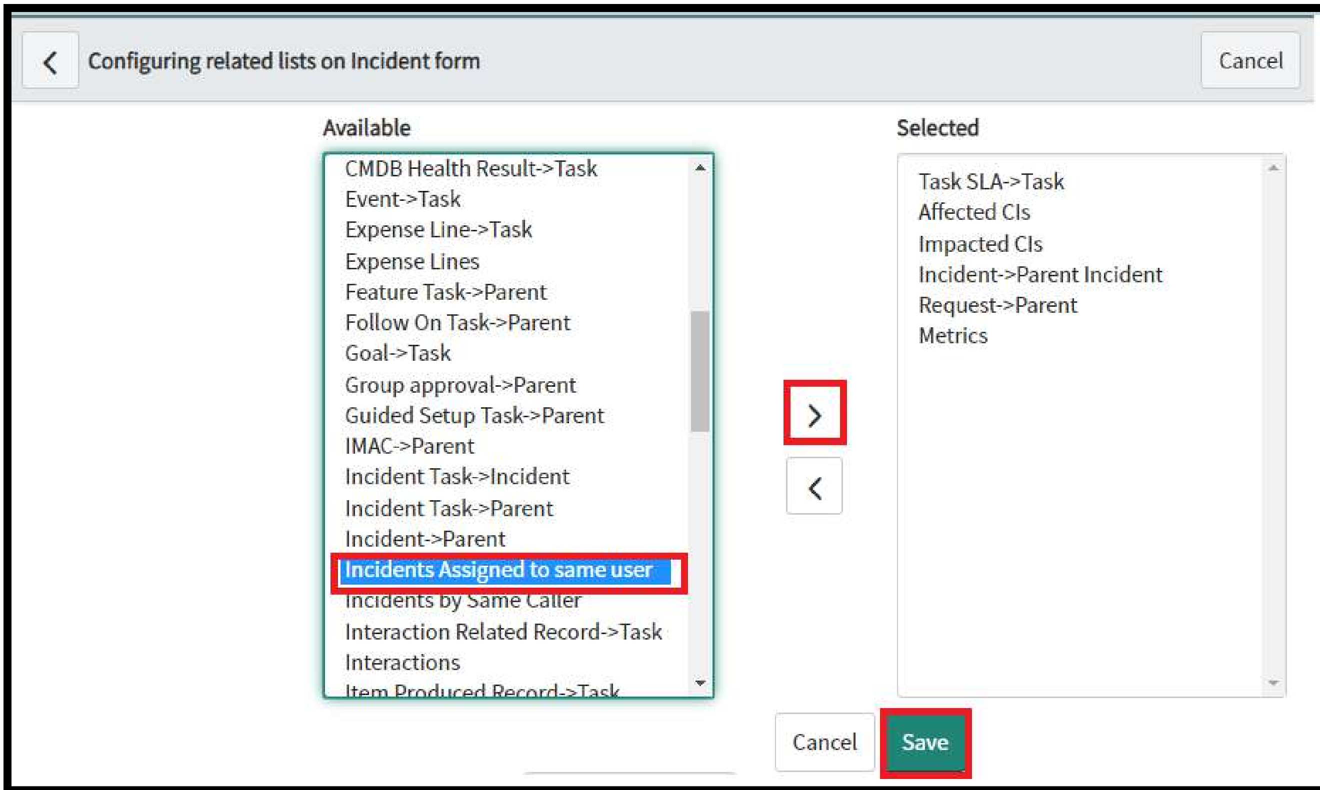
4. **Name:** Incidents Assigned to same user
5. **Applies to table:** Incident
6. **Queries from table:** Incident
7. **Code:** current. addQuery ('assigned_to', parent.assigned_to);
8. Click on **Submit**



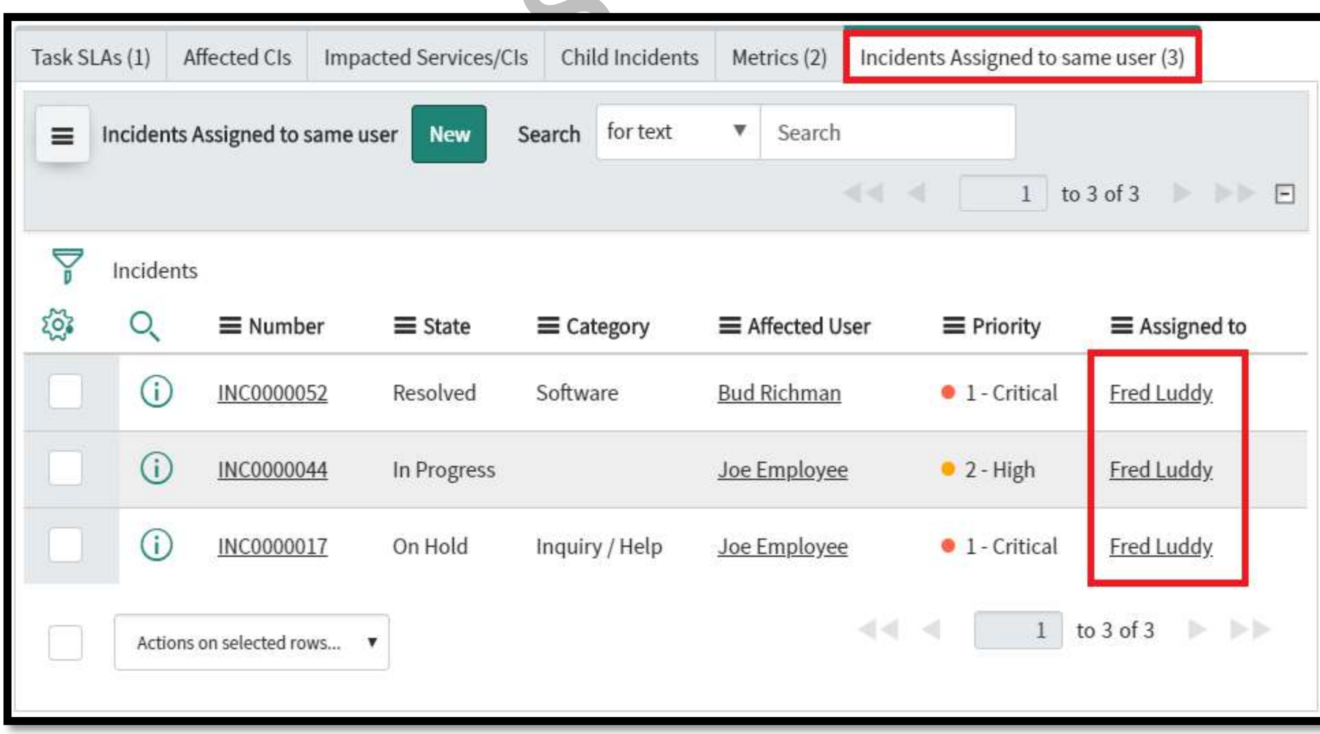
9. Navigate to any existing **Incident** Record
10. Right click on **Form Header**
11. Go to **Configure > Related List**



12. Select **>Incident Assigned to same user** related list
13. Click on **> Add**
14. **Save**



15. Check the result in **Incident record**



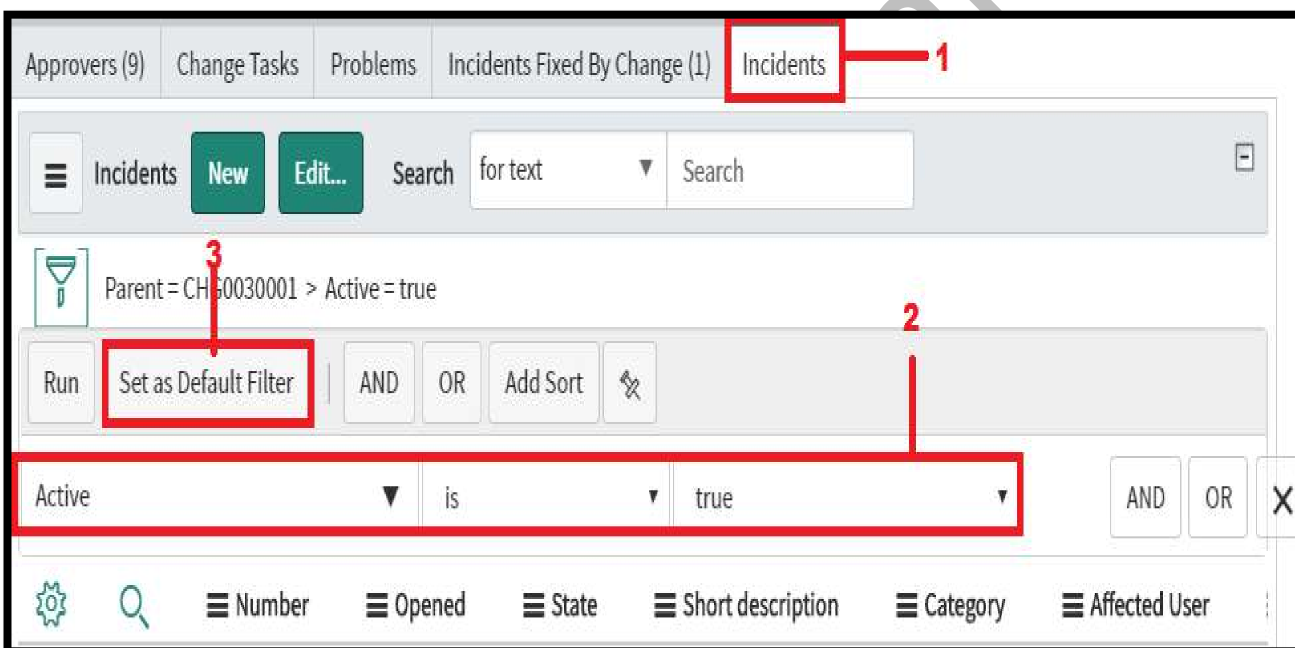
16. If you want explore more into **Related Lists** following existing **Relationship** record

Exercise: 2 Configure default filter for Related Lists

We can create a default filter condition that will working on to a related list when a form loads.

Procedure

1. Navigate to the **Related List >** Incidents record on a **Change** record).
2. Navigate>**Incident** Related List
3. Build the Required filter condition using the **Condition Builder**
4. Condition: **[Active] [is] [true]**
5. Click on **Set as Default Filter**
6. Click on **Run**

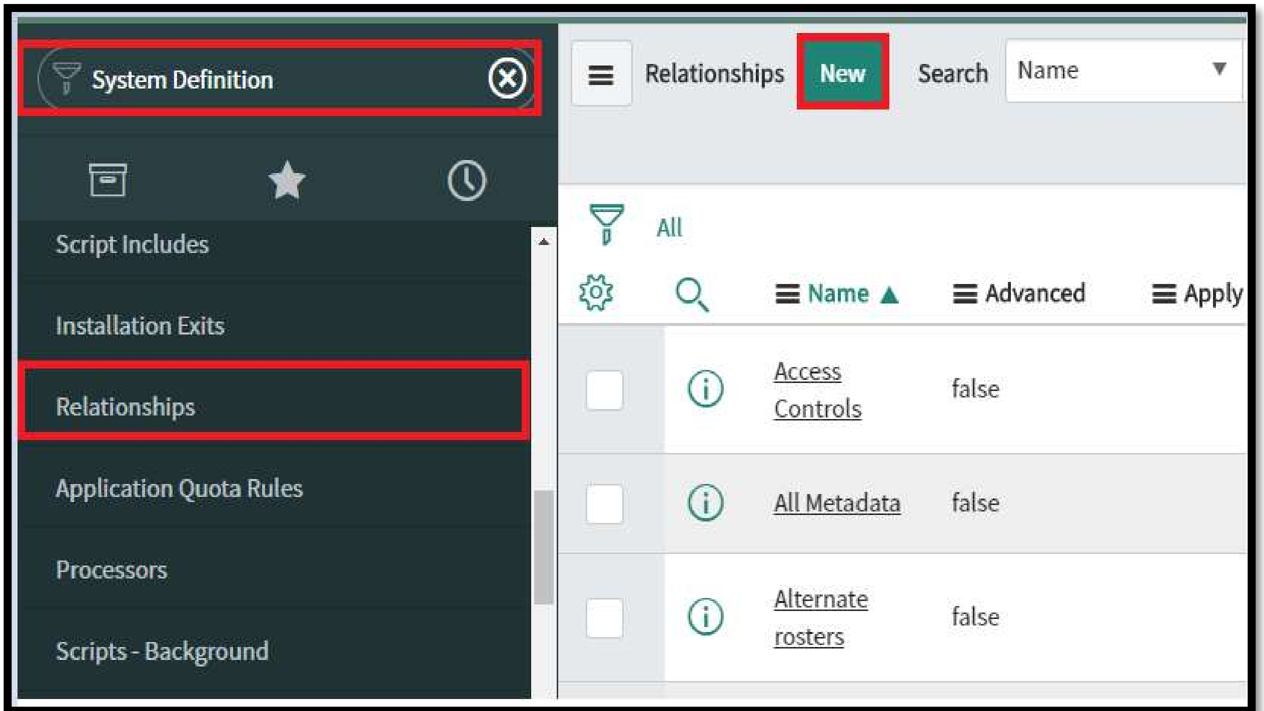


Exercise :4 Configure User Transaction Related List

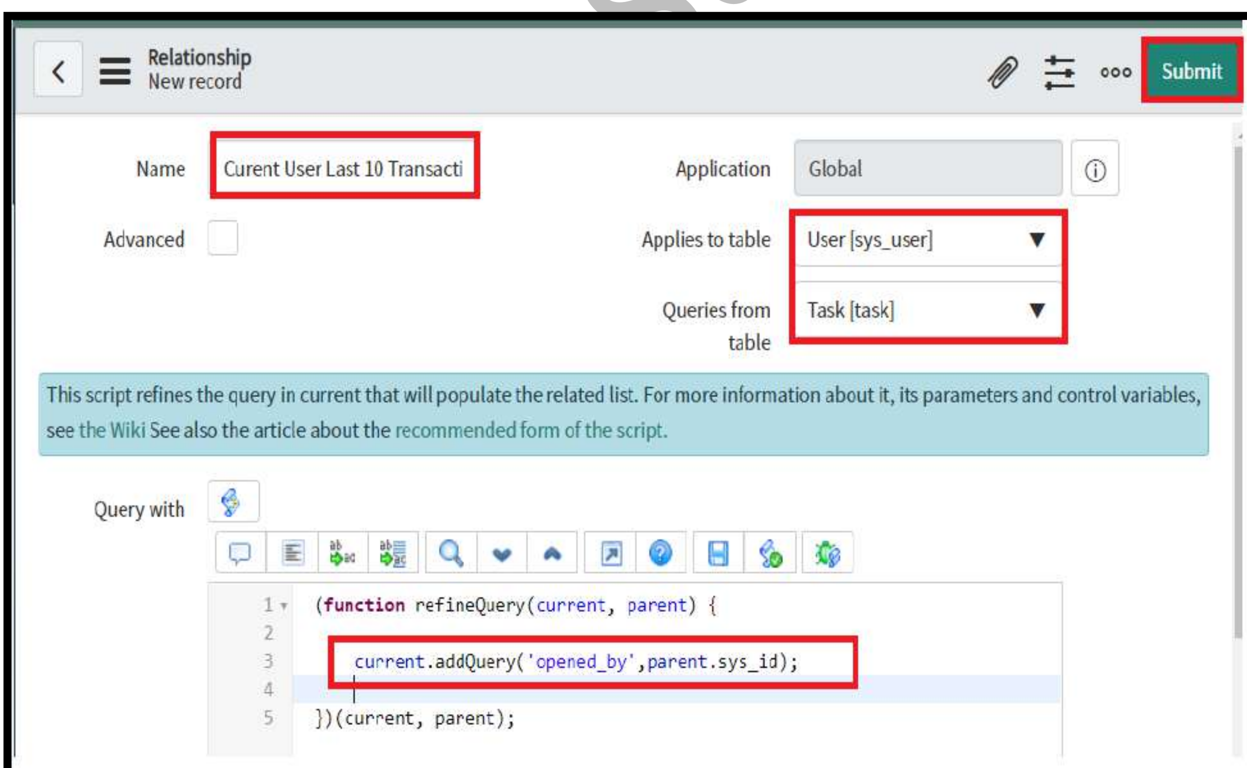
This Relationship will show last 10 transactions performed by current user

Procedure

1. Navigate to **System Definition > Relationships.**
2. Click on **New.**

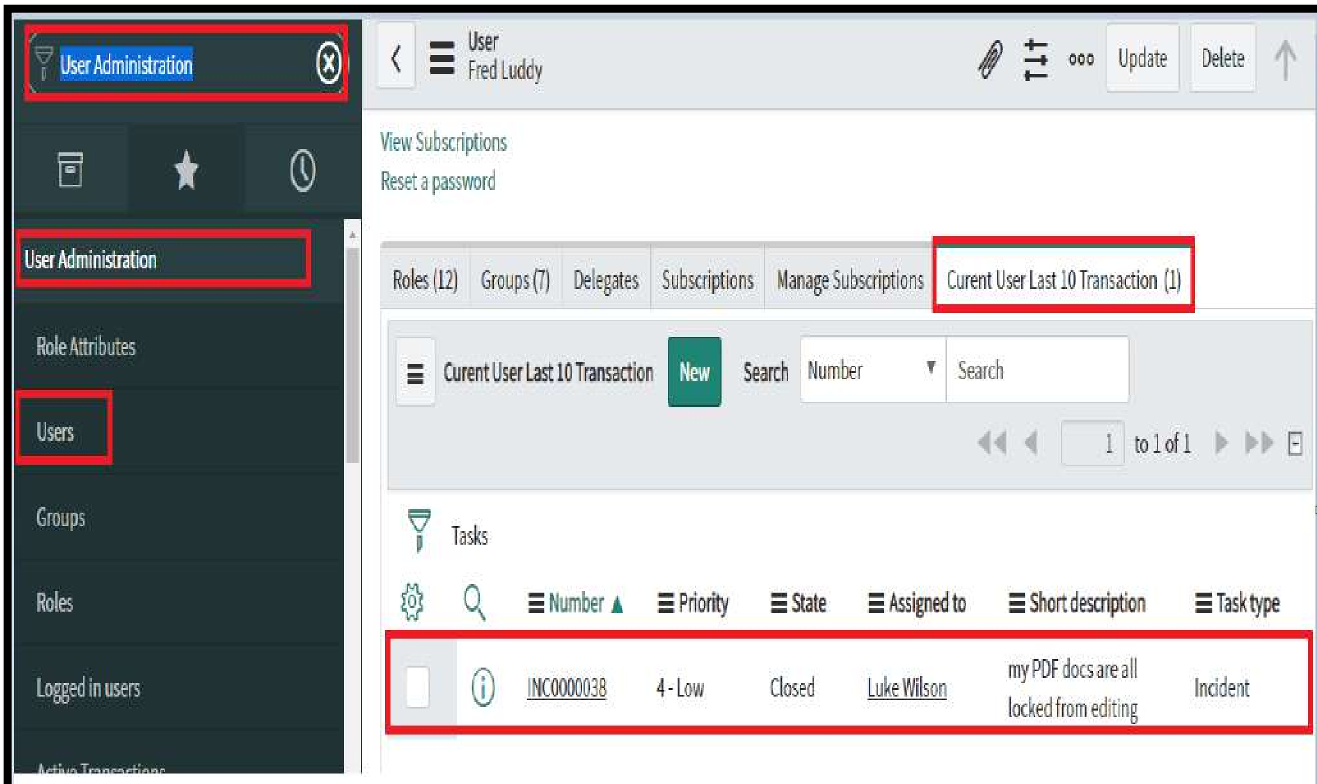


3. Fill **Relationship** Form



4. **Name:** Current user last 10 transaction
5. **Applies to table**: User[sys_user]

6. **Queries from table: Task**[Task]
7. **Code:** current.addQuery('opened_by' , parent.sys_id);
8. Click on **Submit**
9. Navigate to > **User Administration > Users**
10. Open any particular **User Record (Ex. Fred Luddy)**
11. Right Click on **Form Header**
12. Add **Current User Last 10 Transaction** Related List to your form
13. Check **Result** like below



Chapter – 13

SLA Definitions

Define Service Level Management

In any business service the Service Level Managers are responsible for a set of agreements between a service provider and customer that define the **scope, quality and speed** of the services being provided.

Service level agreements **(SLAs)** provide the customer with an expectation of service within a known time line and the capability to monitor when expected service levels are not being met.

Purpose of SLA

We can use the SLA Definition document to define a specific set of business criteria that would result in an SLA being running.

We need to define some of the following parameters:

- ✓ **Table**: SLAs can be defined against any particular table like. **Incident**
- ✓ **Duration:** You can specify the time duration in which the service must be provided to the customer.
- ✓ **Schedule:** We can define the schedule time, which represent valid working and not-working days that the service provider follows to deliver the service. The selected schedule will be used to determine when the SLA will breach.
- ✓ **Conditions:** You can specify the conditions under which the SLA will **start, pause, stop, or reset.**

When any SLA Definition is triggered against a particular task the Task SLA record is created and contains all the tracking data for the specific SLA on that record.

If an SLA Definition exists for **Priority 1** incidents a Task SLA record **[task_sla]** will attach to the P1 incident record and capture all the data associated with it. In many cases there will be multiple Task SLA records against a single task since many definitions apply.

Objectives of SLA

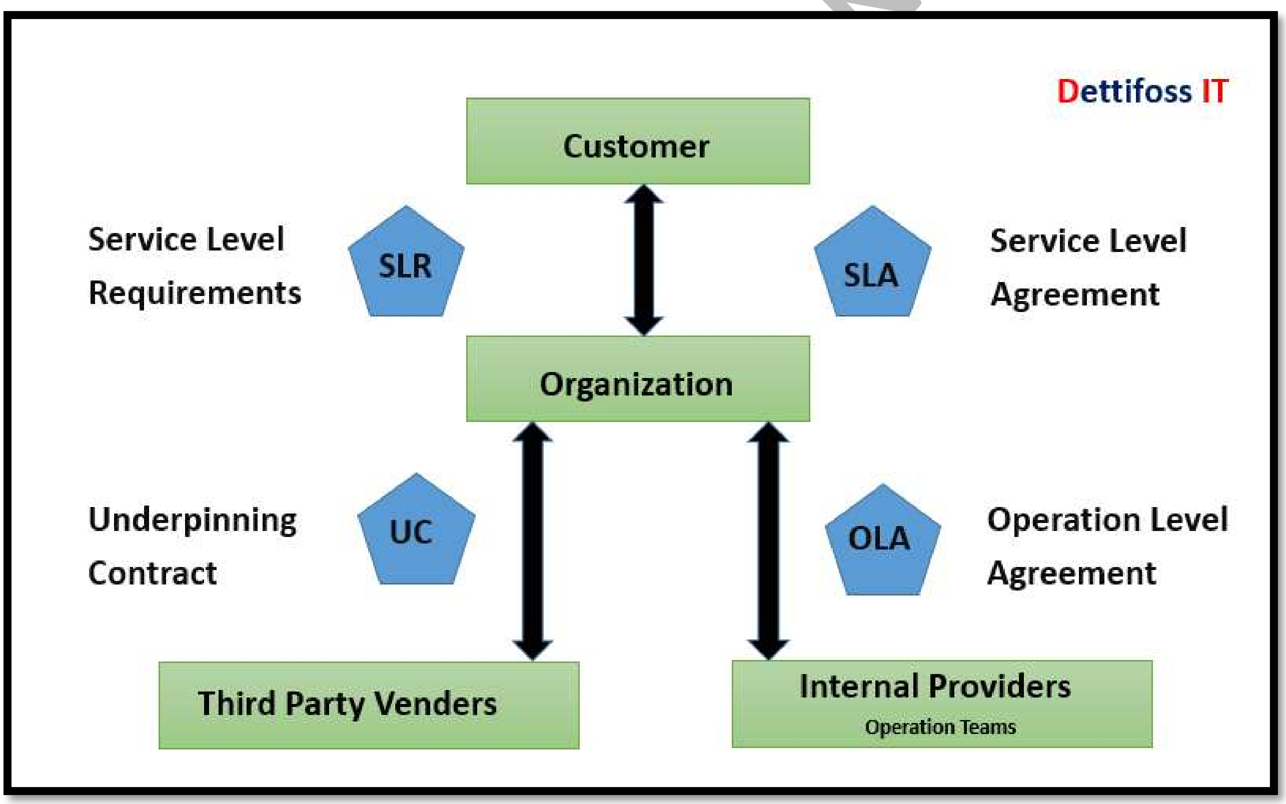
- ✓ Define, Document, Agree, Monitor, Measure, Report and Review the level of IT Service Provided
- ✓ Provide and improve the relationship and communication with the business and customer
- ✓ Ensure the specific and measurable targets are developed for all IT service
- ✓ Monitor and improve the customer satisfaction with quality of service delivered

Create new SLA definition

An SLA definition record can define the **conditions, schedule, duration, timings, workflows,** and other information required to create and progress **task SLAs**. We can able to create **one or more SLA** definition records based on our client requirement

Priority 1 resolution time is (1 hour) SLA Definition defines the **Task SLAs** to attach to incidents with a P1 - Critical priority, specifies appropriate conditions for those Task SLAs, and uses the default SLA workflow to create events such as to send a notification, when an incident's Task SLA reaches **50%** of its allotted time and it can provide information about breaches

SLM Architecture – Agreements



Types of SLAs (SLA, OLA, UC)

Service Level Requirement (SLR)

A set target and responsibilities documented agreed upon for each proposed new or changed services. SLR based upon business objectives and used to negotiate agreed upon service level targets

Service Level Agreement (SLA)

An agreement between an It service provider and IT customer, it defines key service target and responsibilities of both parties.

Operation Level Agreement (OLA)

An agreement between an IT service provider and another part of same organization that assist with the provision of service. A facilities department that maintaining.

Network team supports the (network services). An OLA should contain targets

Underpinning Contract (UC)

A contract between an IT service provider and external third part vendor covering delivery of services that support it organization in its delivery of services.

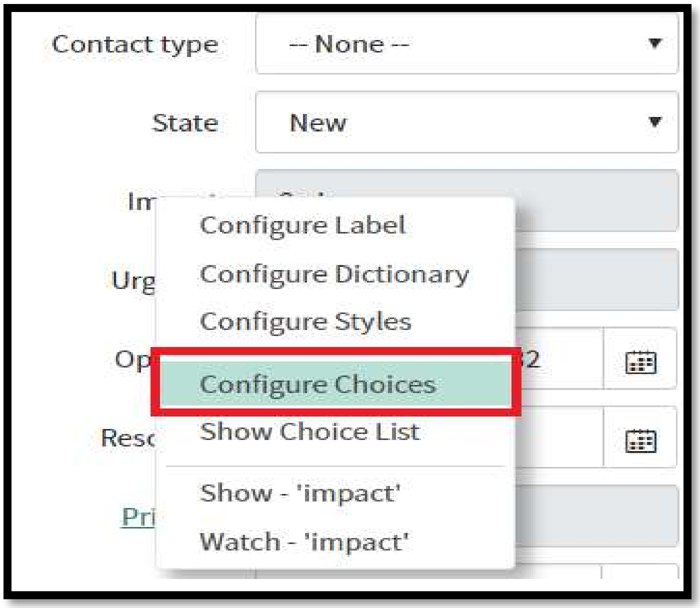
Exercise: 1 Add the choices to below fields

First we have to add choices for **Impact, Urgency, Priority** fields

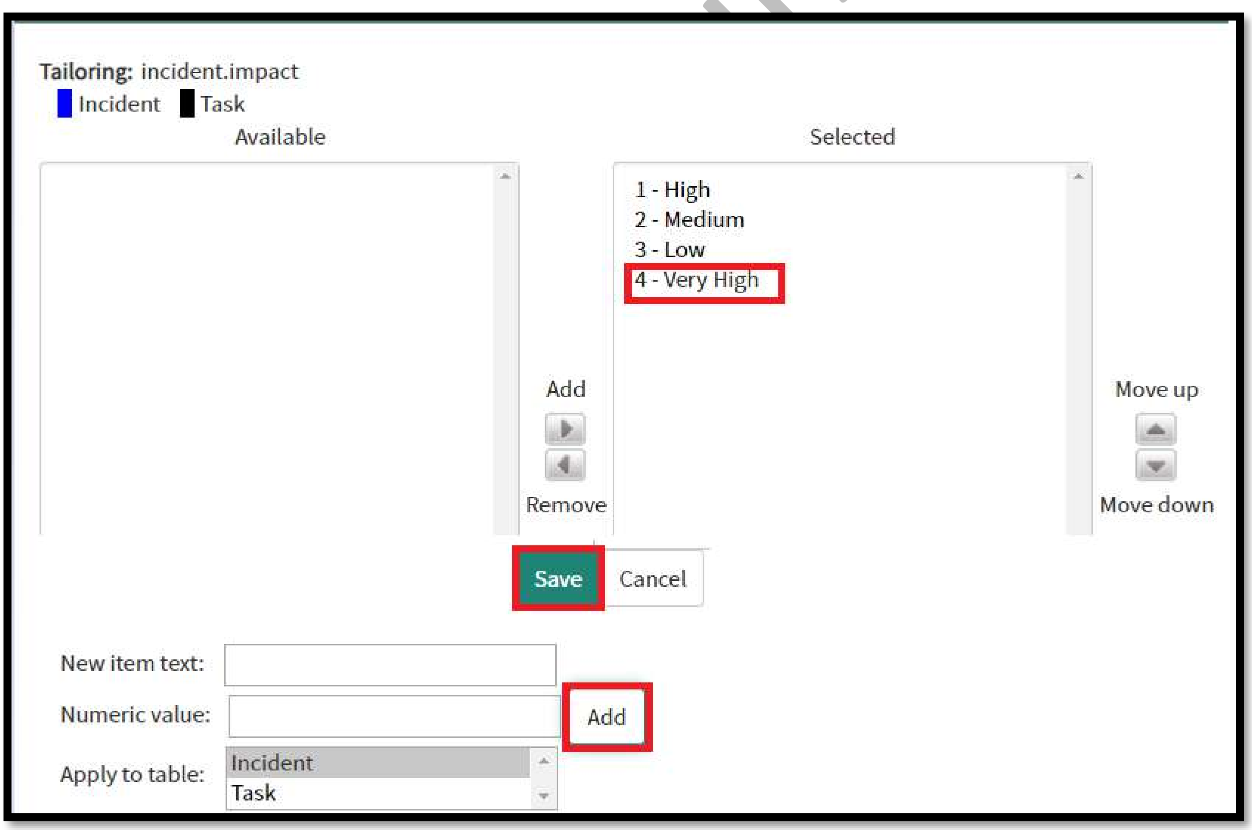
Impact: 4 –Very High, **Urgency:** 4 – Very High, **Priority:** 6 – Very Critical

Procedure

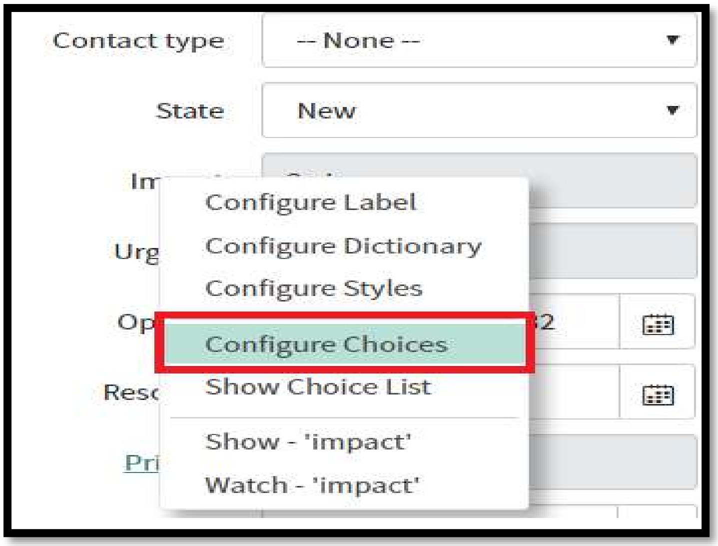
1. Right click on **Impact** label
2. Click on **Configure Choices**



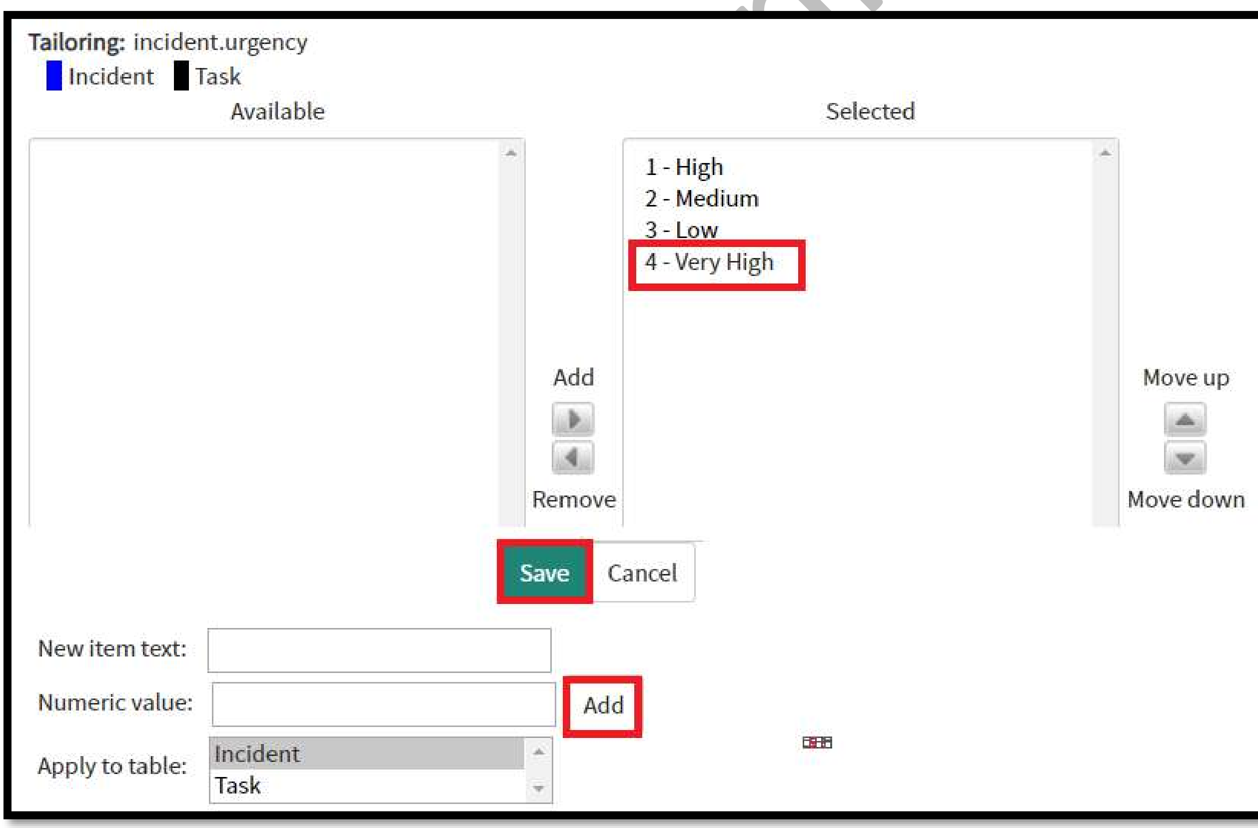
3. Add the choice **4 – Very High**



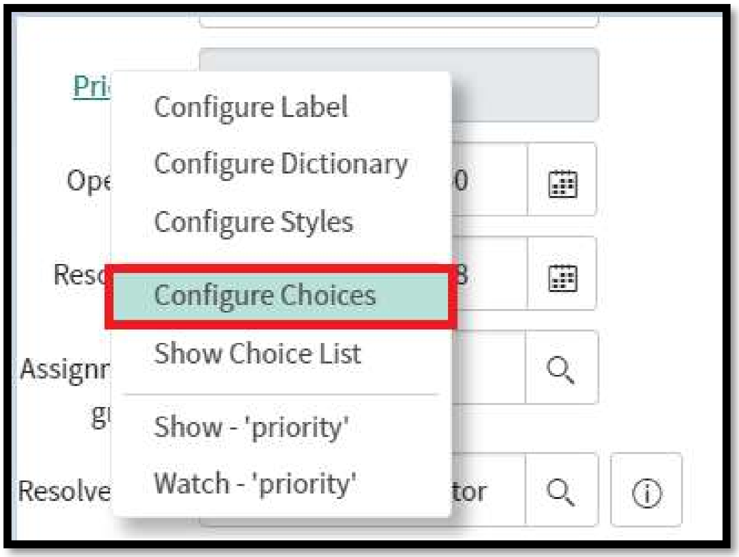
4. Click on **Save**
5. Right click on **Urgency** label
6. Click on **Configure Choices**



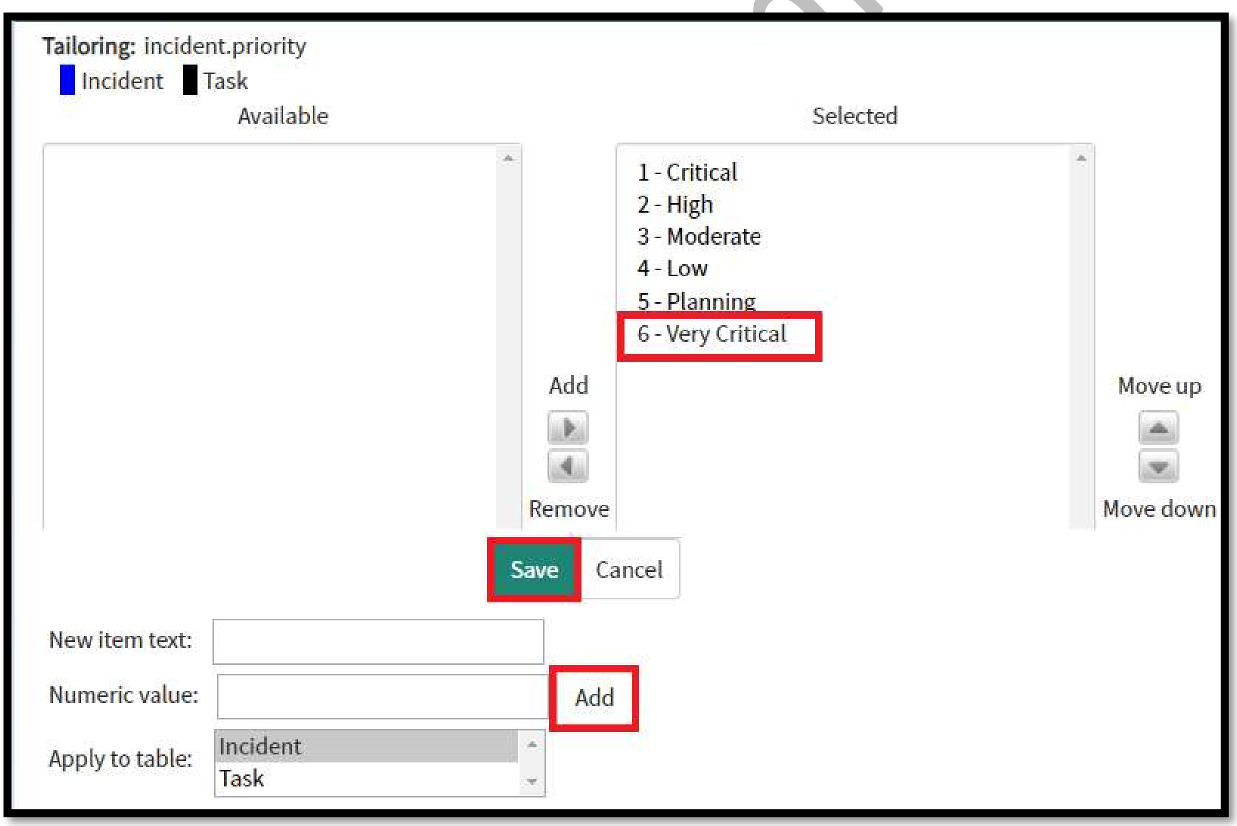
7. Add the choice **4 – Very High**



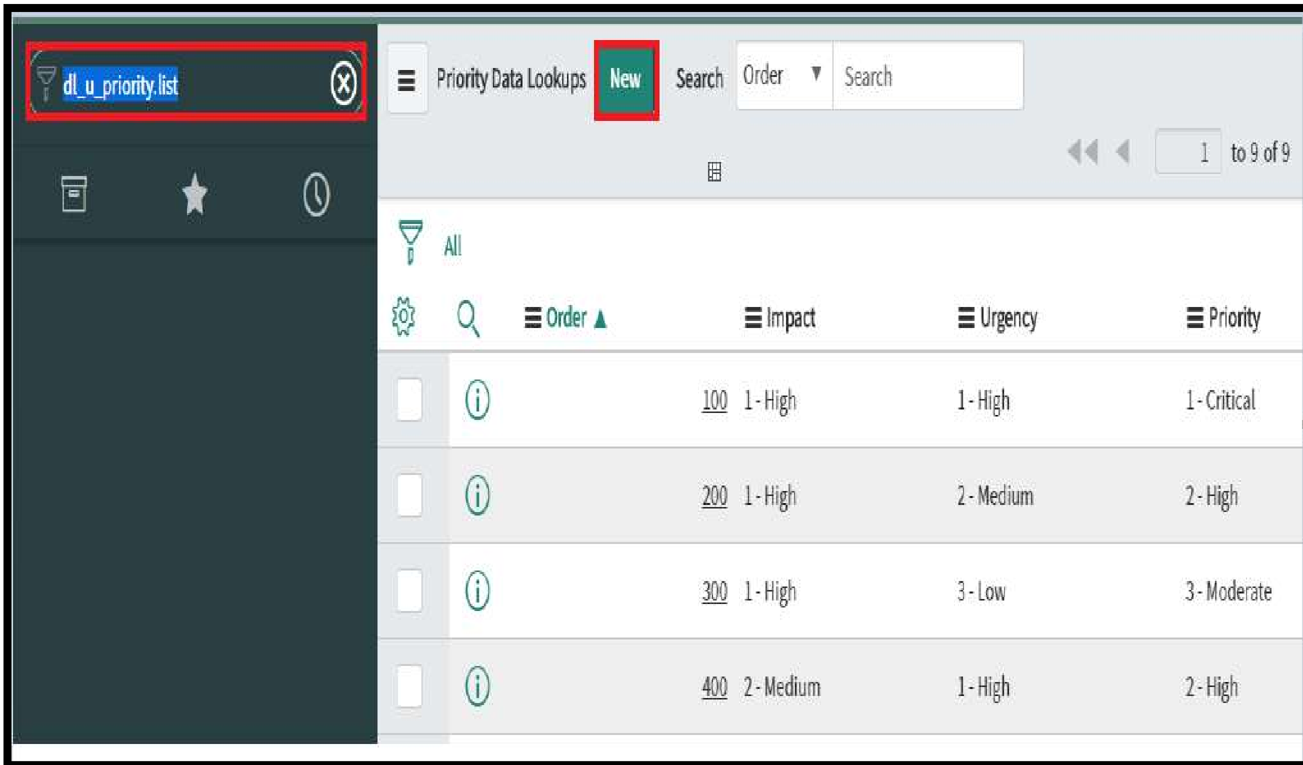
8. Click on **Save**
9. Right click on **Impact** label
10. Click on **Configure Choices**



11. Add the choice **6 – Very Critical**

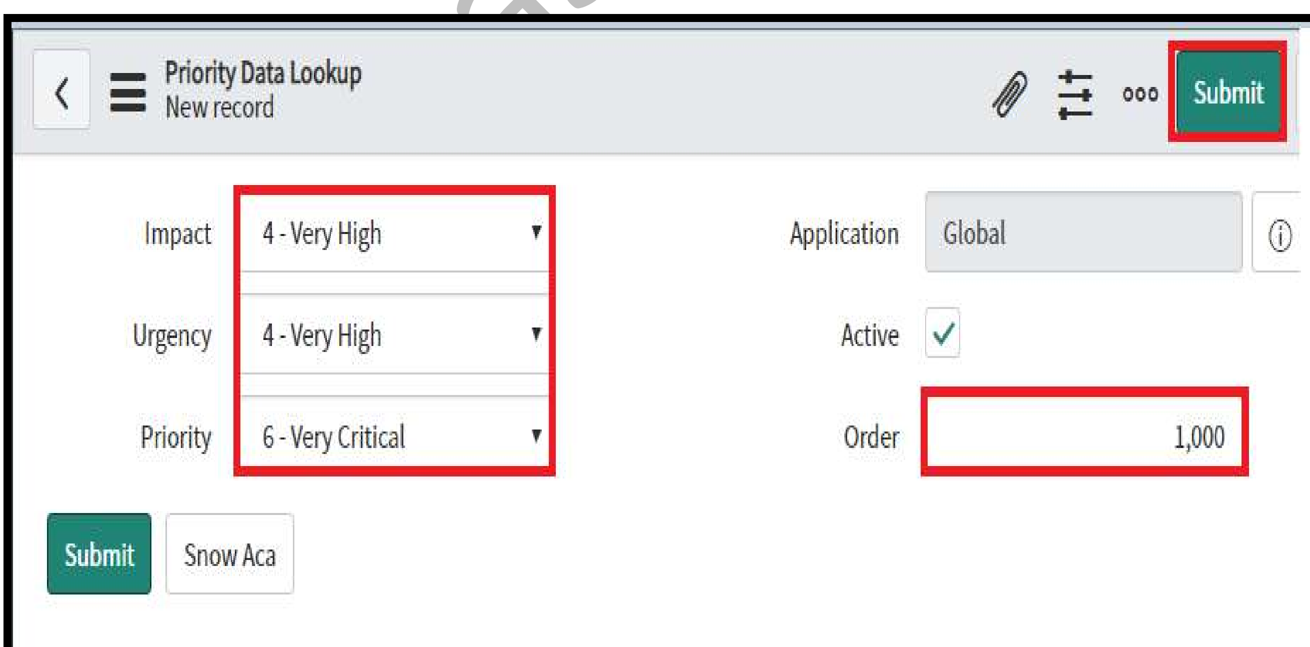


12. Click on **Save**
13. Configure **Lookup Definition** Record in **dl_u_priority** table
14. Navigate to filter navigation open table **dl_u_priority.list**



15. Click Priority Data Lookup **New**

16. Fill the below form



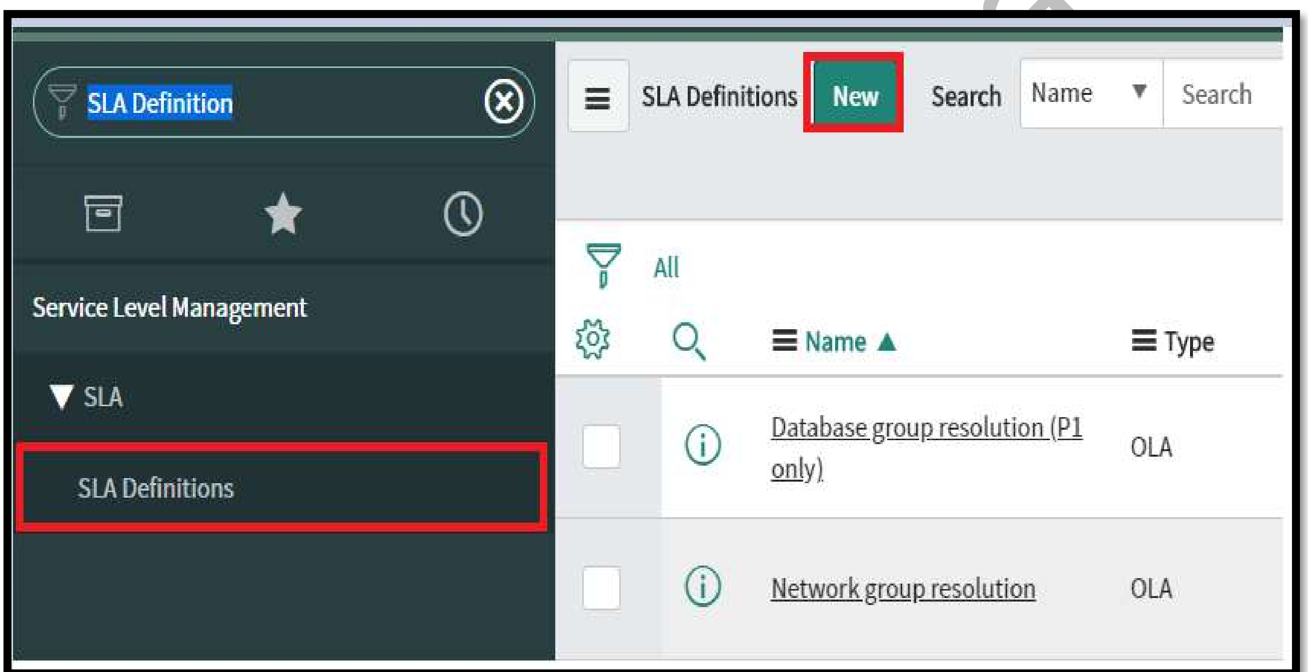
17. Click on **Submit**

Exercise: 2 Create new SLA definition

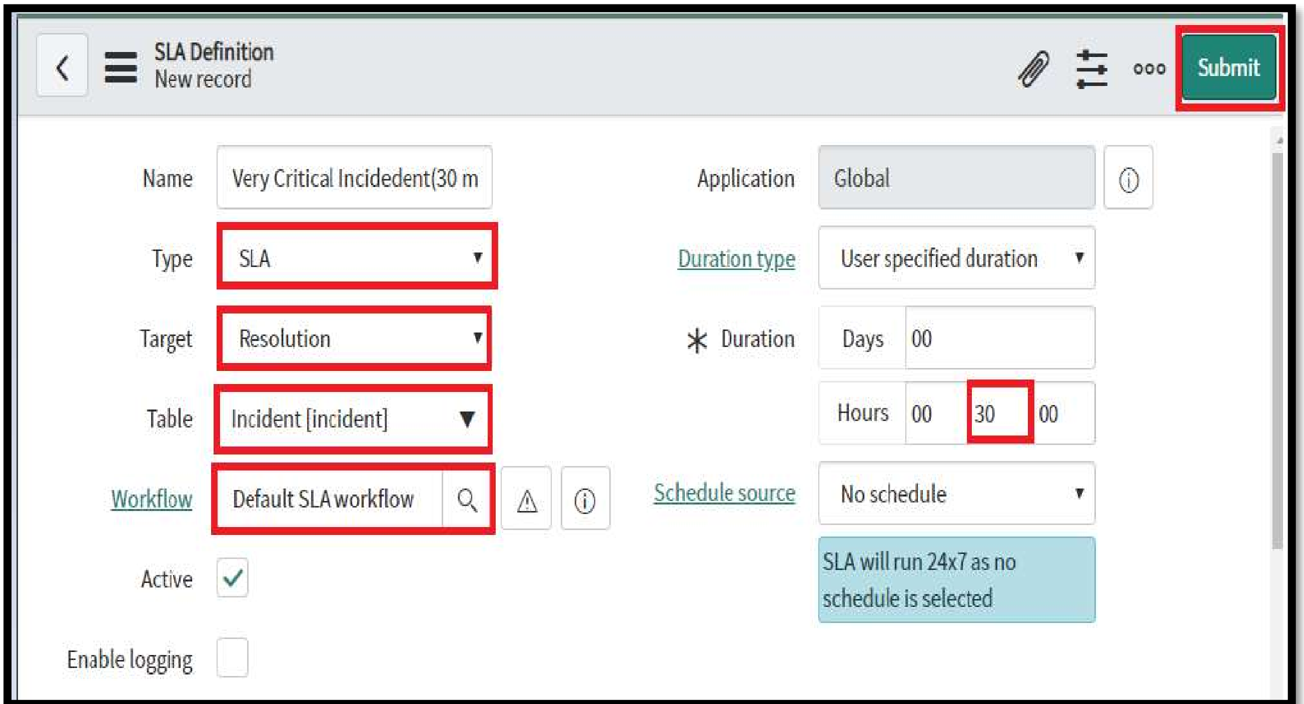
Any customer created a **Very Critical Priority** Incident Then This SLA Definition record will attach to that incident

Procedure

1. Navigate to **Service Level Management > SLA Definition**
2. Click on **New**
3. Fill the below details

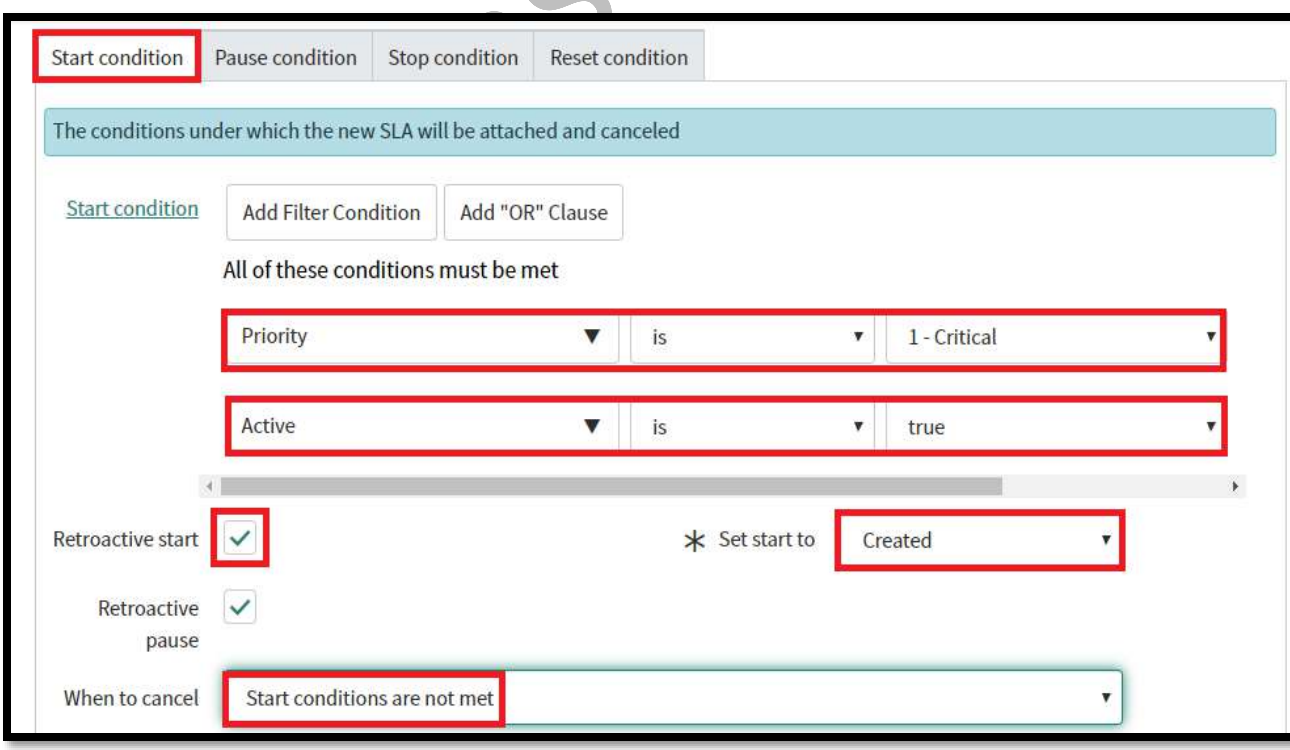


4. **Name** : Very Critical Incident (30 mins)
5. **Type** : SLA
6. **Target** : Resolution
7. **Table** : Incident
8. **Workflow** : Default SLA Workflow
9. **Duration**
10. **Schedule Source:** No Schedule

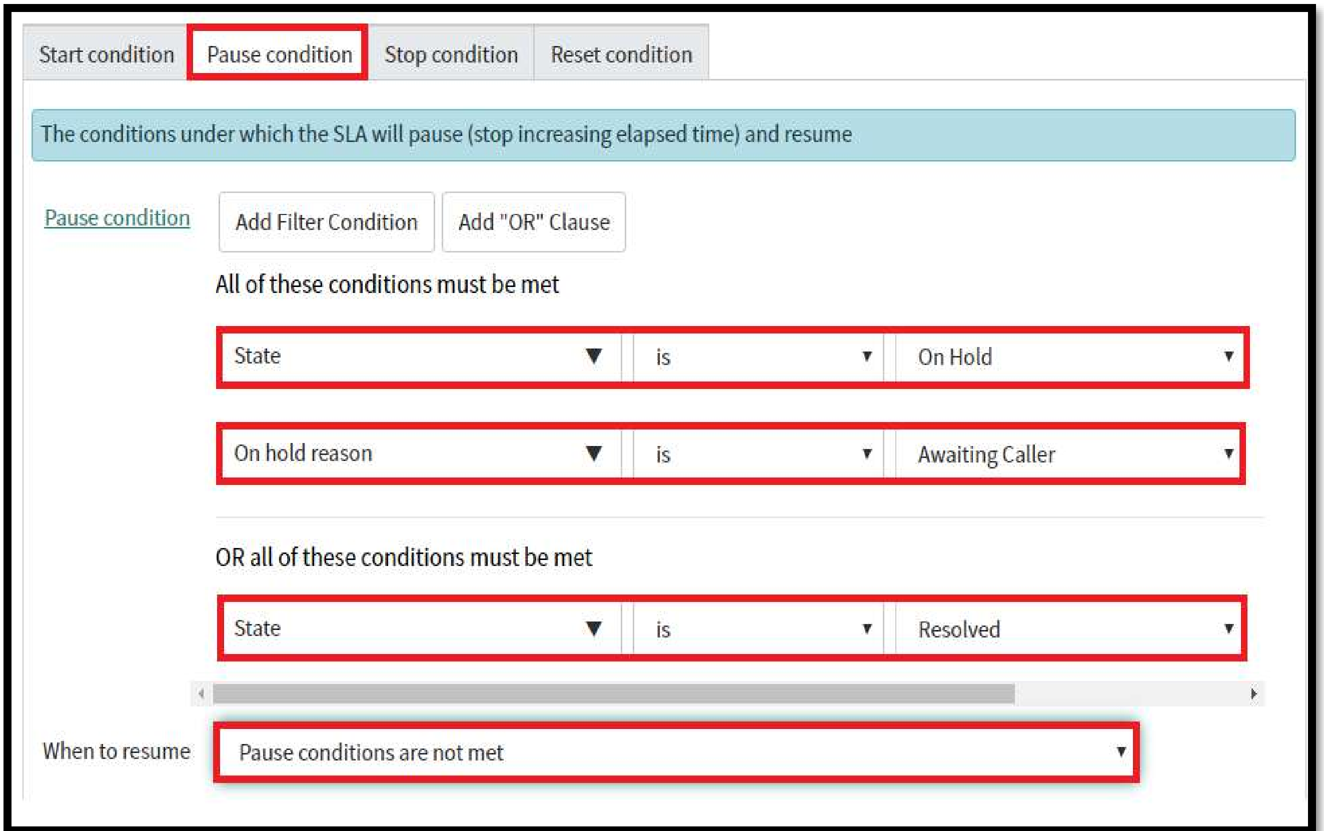


11. Apply the conditions when it has to be **start, pause, stop, reset**

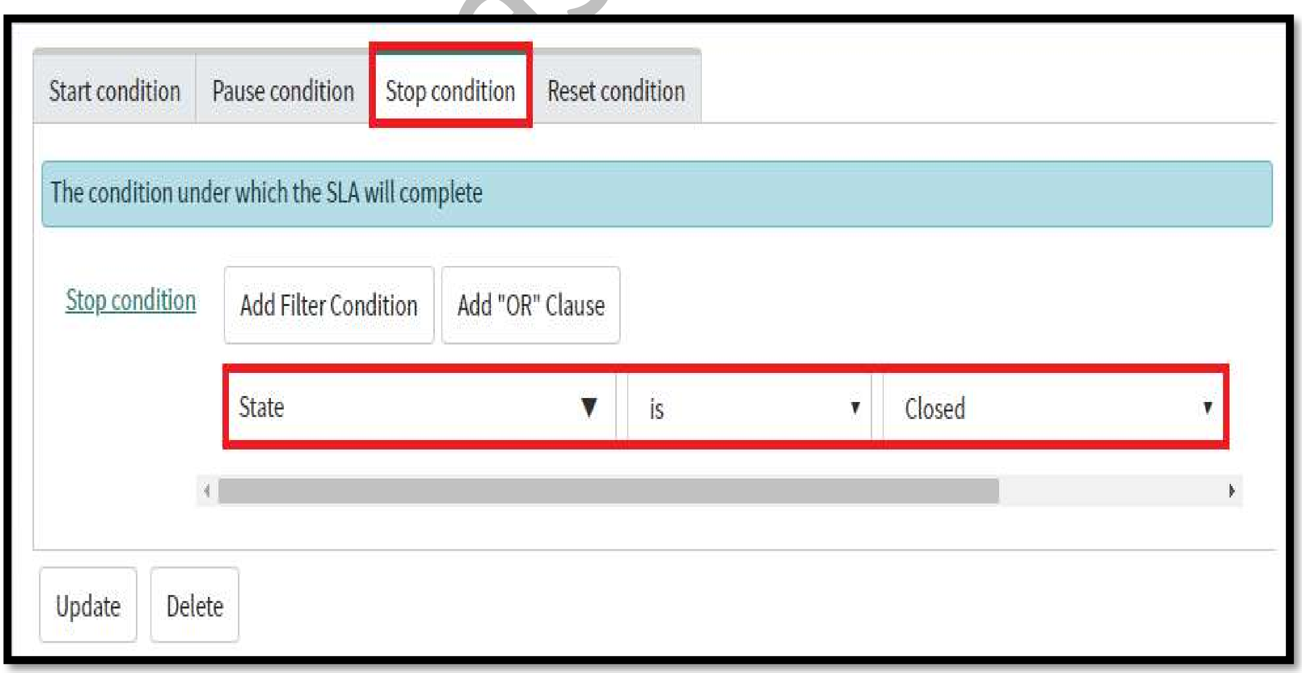
12. Start Condition



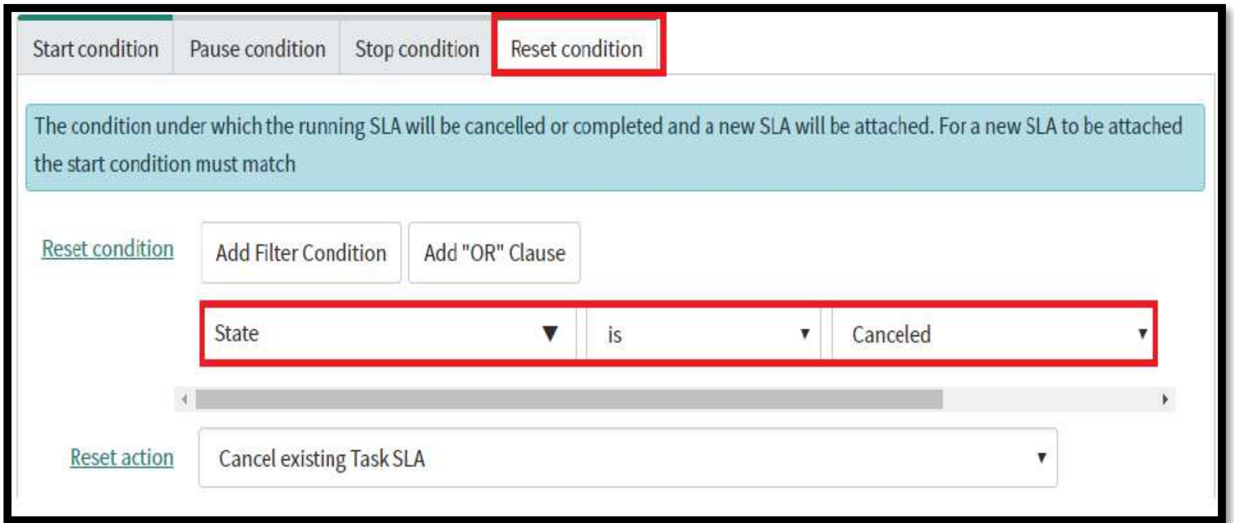
13. Pause Condition



14. Stop Condition



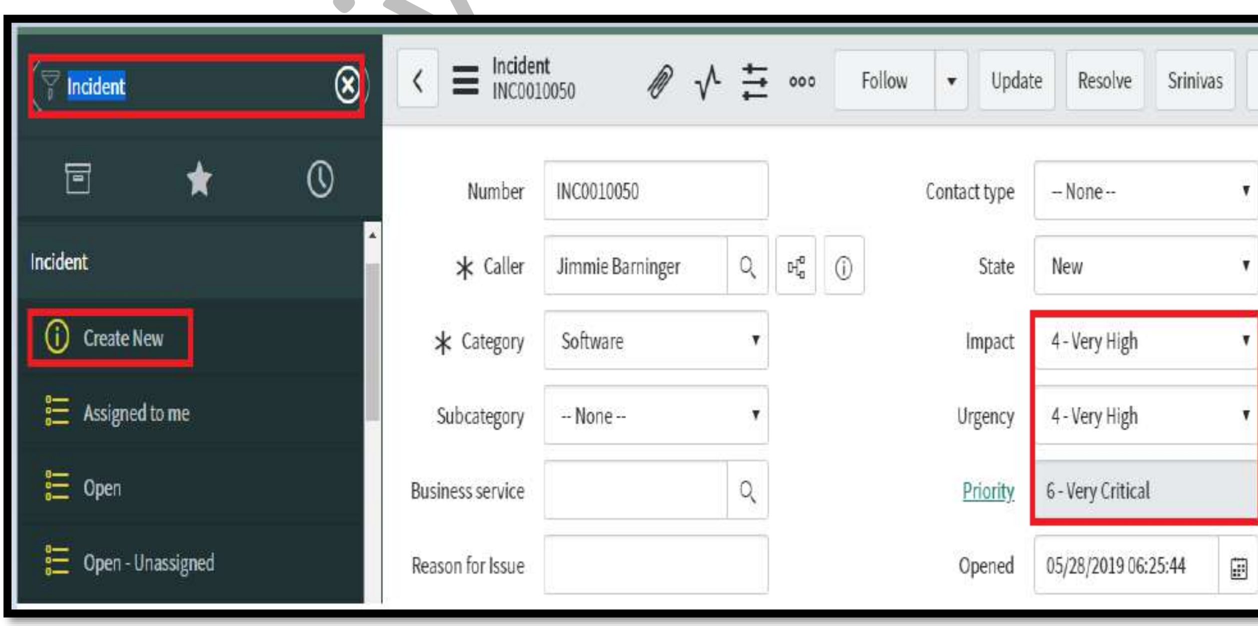
15. Reset Condition



16. Click on **Submit**

SLA Task: 1

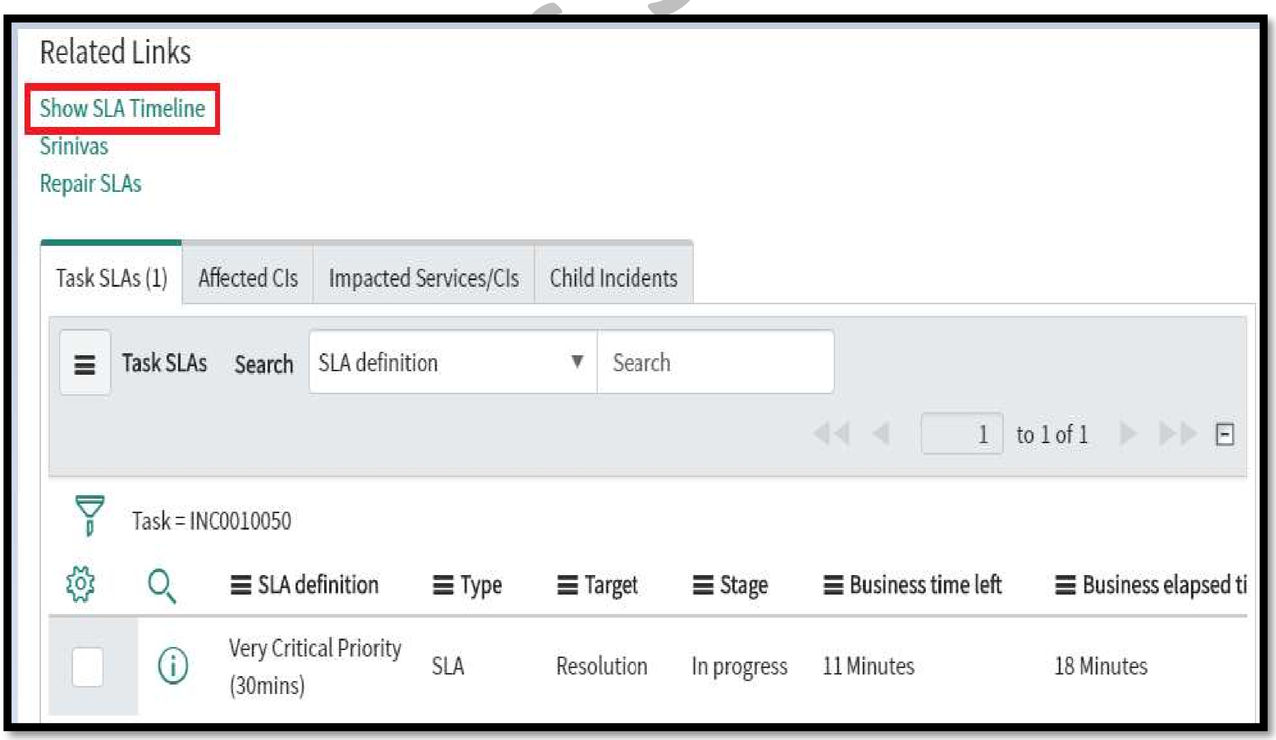
1. Navigate to **Incident Application > Click on New Module**
2. **Impact** : Very High
3. **Urgency** : Very High
4. **Priority** : Auto filled (**Very Critical)**
5. Click on **Submit**

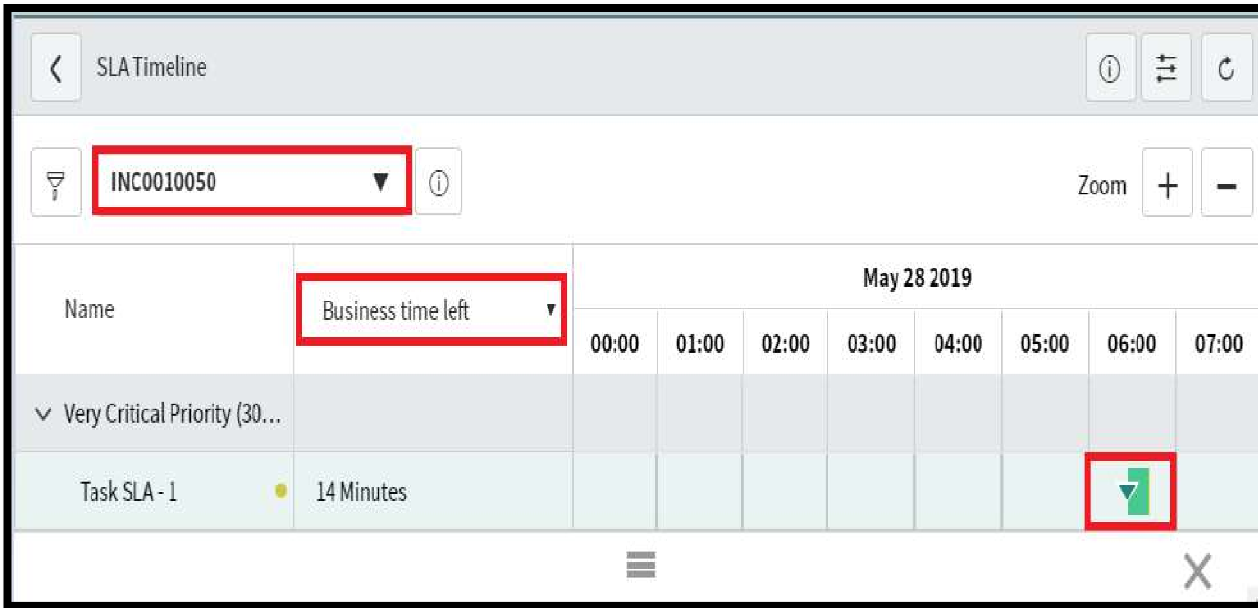


6. **Again** open same record

7. Navigate to **Task SLA** from Related list

8. This example demonstrates how an SLA can be attached to an incident, then progressed to completion.
9. **SLA Time Line**





SLA Calculation for exact time

As a result, breaches can occur when SLA calculations are calculated up to 100%. You can enable SLA calculations to instead use the business time or actual time left.

For example, a business percentage of 99.951% is rounded up to 100%, which causes a breach.

For more accurate SLA calculations, enable the SLA engine property for using the exact value of the business time left if a schedule is specified or the actual time left if the SLA has no schedule specified.

Chapter – 13

Import Sets

What is import sets?

Import sets will allow service now administrators to import the data from various external data sources, and then we can map that data into Service Now target tables.

1. The Import Sets table can act as a staging area for records imported from various data source
2. Data should not be imported in extremely large files that may cause some problem. Creating an extremely large import set can cause delays and system outages.
3. A transform map can be defining the relationships between fields displaying in an Import Set table and fields in an existing Service Now table
4. Importing sets will skip records when the data in the instance matches the data being imported.
5. Import Sets can run on as user System. Therefore, Import Sets cannot add data to encrypted fields

Import Sets Supporting Files

CSV	This files that double quotes may not appear inside fields. The first row of data in an imported CSV file becomes the header row and defines the columns for that import set.
Excel	Excel files should have the extension of XLS or XLSX extension.
XML	XML files should have a consistent XPath for each data row from this file.
JSON	For JSON arrays, the path for each row must specify the array root element twice, such as

Supporting External Data Sources

JDBC	Importing data through some JDBC connection require Mid Server Configuration
LDAP	LDAP imports require a valid transform map.

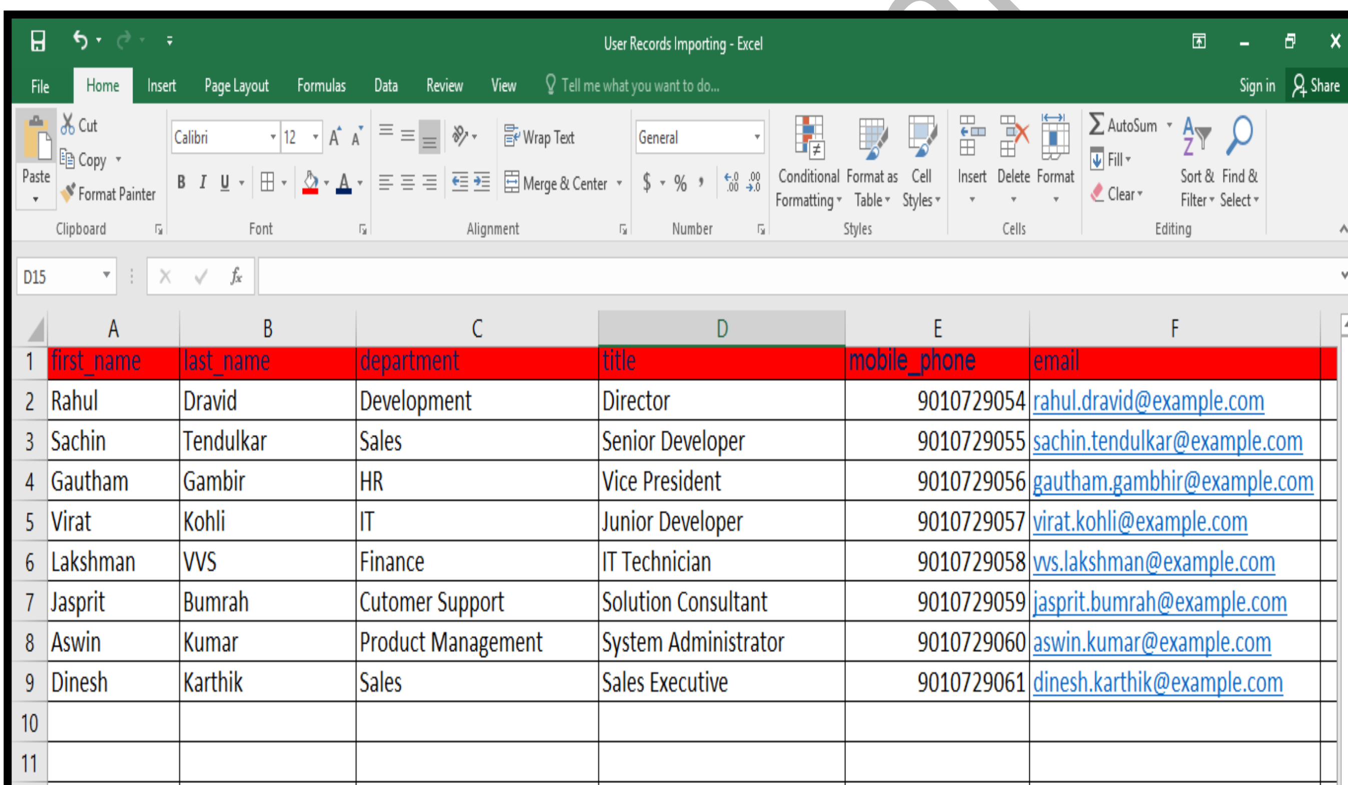
Exercise: 1 Importing Excel File

We will import **Excel File** that is representing each row is record in a file
Following below steps to importing data in service now tables.

We are going to importing the data into **sys_user** table

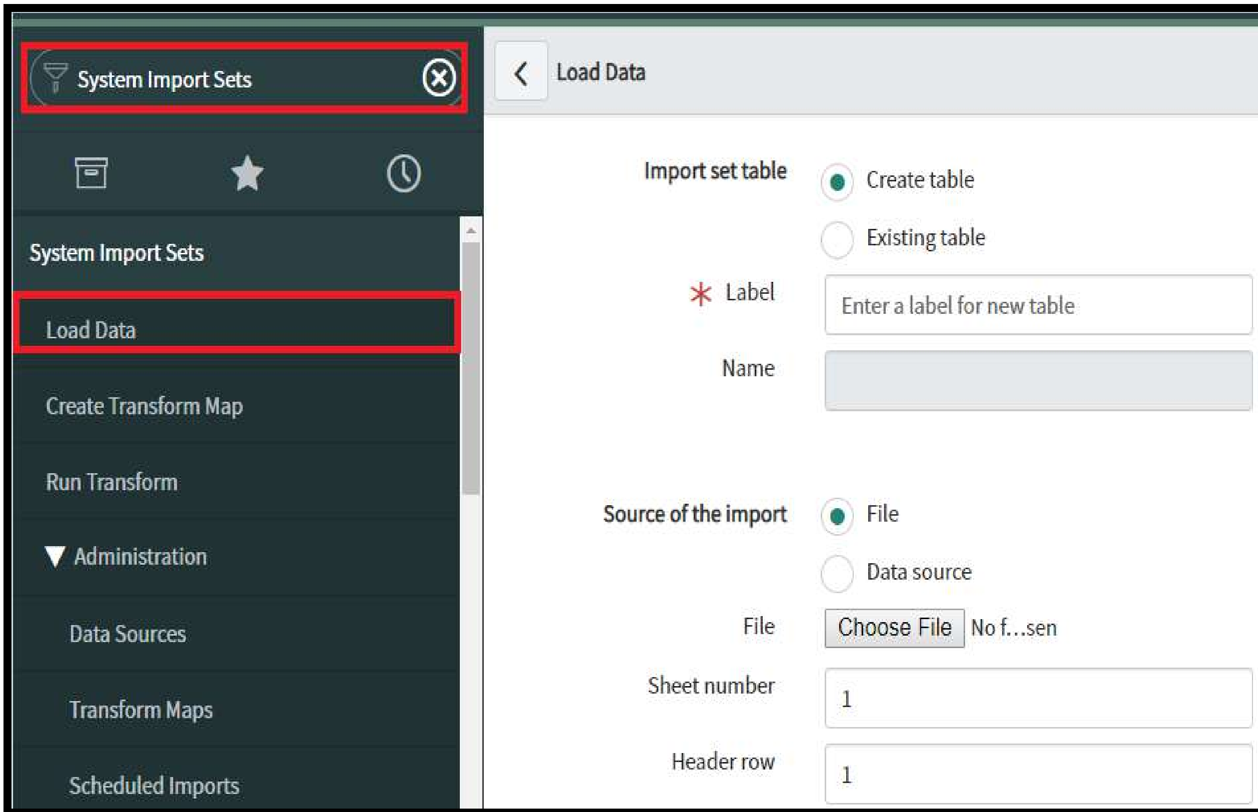
Procedure

1. First we need to create **Data Source File**
2. Open Empty **Excel File** and Crete **8 or More** user records

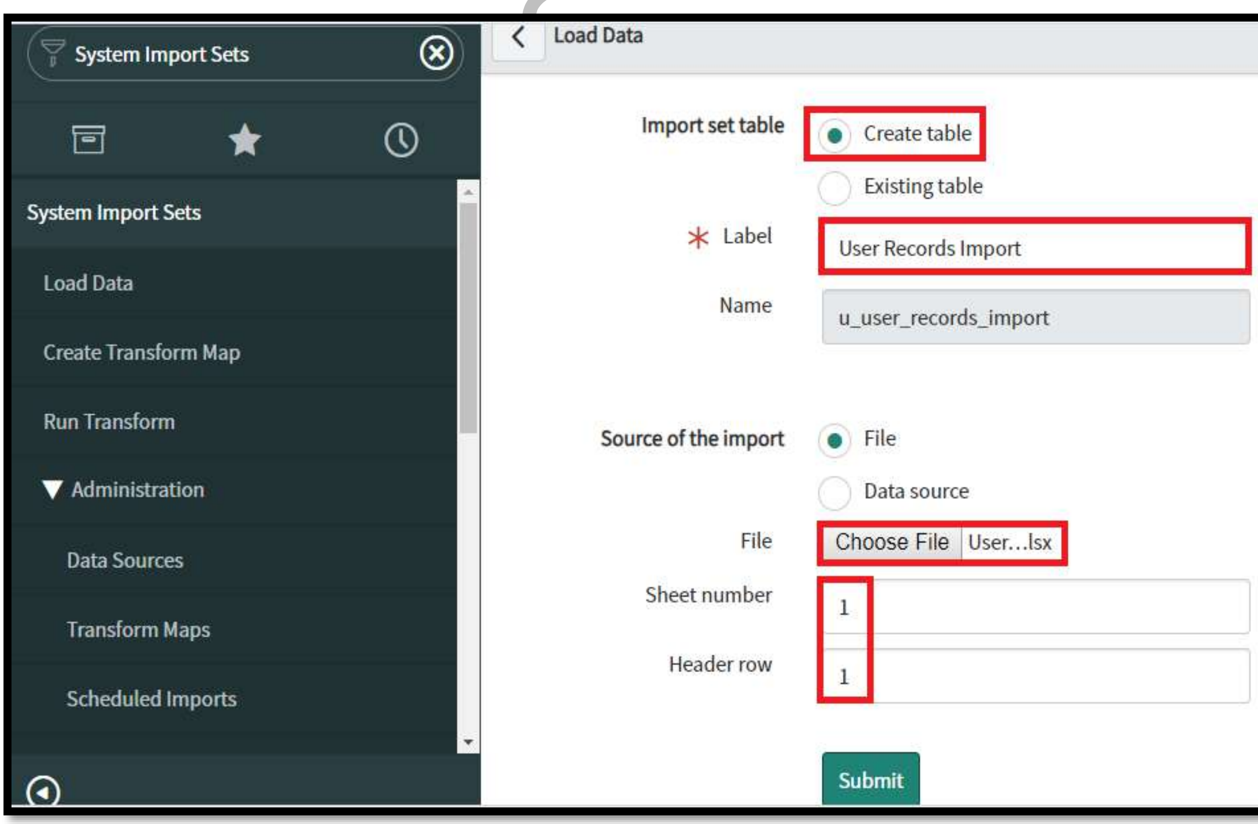


	A	B	C	D	E	F
1	first_name	last_name	department	title	mobile_phone	email
2	Rahul	Dravid	Development	Director	9010729054	rahul.dravid@example.com
3	Sachin	Tendulkar	Sales	Senior Developer	9010729055	sachin.tendulkar@example.com
4	Gautham	Gambir	HR	Vice President	9010729056	gautham.gambhir@example.com
5	Virat	Kohli	IT	Junior Developer	9010729057	virat.kohli@example.com
6	Lakshman	VVS	Finance	IT Technician	9010729058	vvs.lakshman@example.com
7	Jasprit	Bumrah	Cutomer Support	Solution Consultant	9010729059	jasprit.bumrah@example.com
8	Aswin	Kumar	Product Management	System Administrator	9010729060	aswin.kumar@example.com
9	Dinesh	Karthik	Sales	Sales Executive	9010729061	dinesh.karthik@example.com
10						
11						

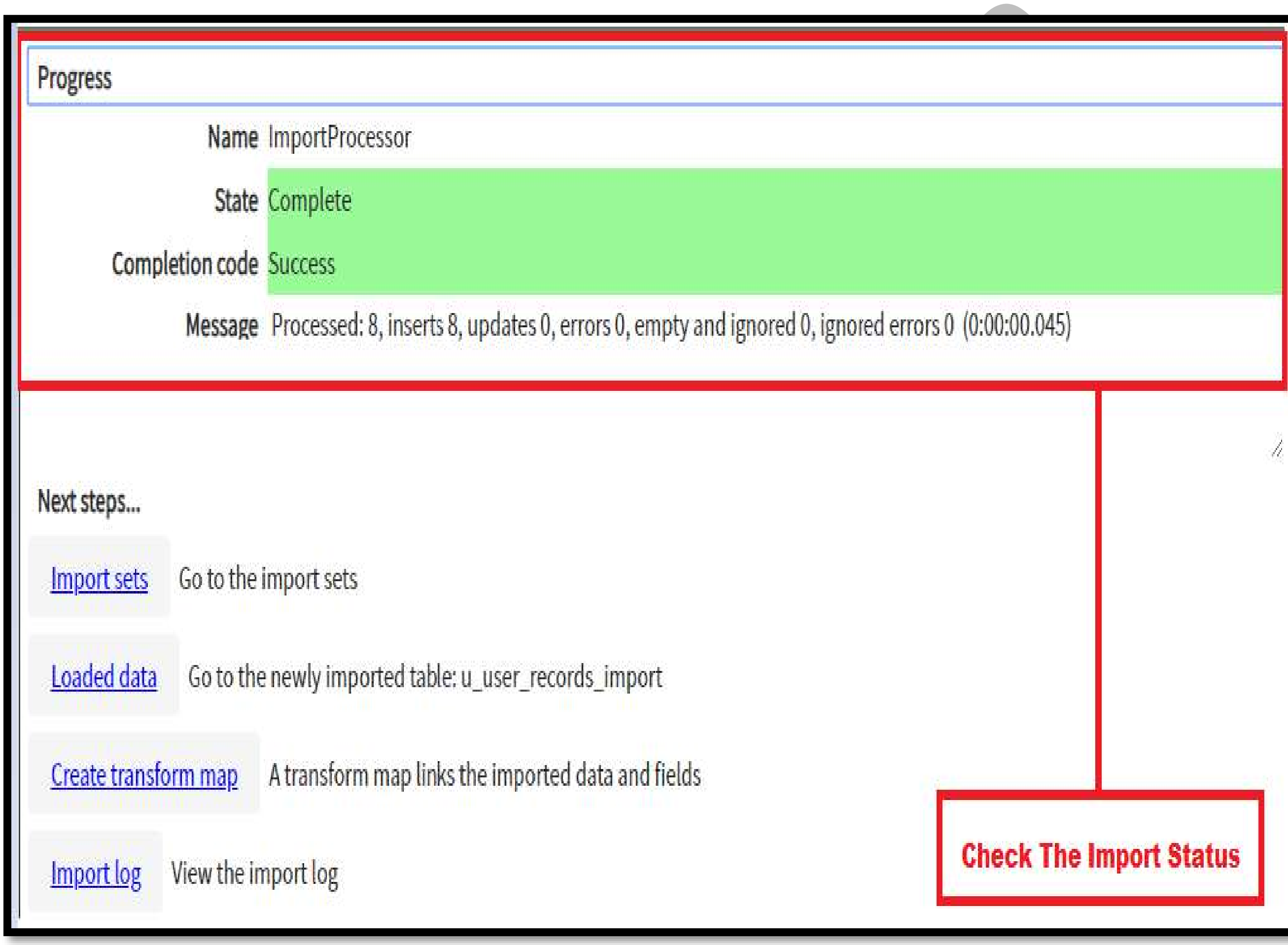
3. Navigate to **System Import Sets**
4. Click on **Load Data** module



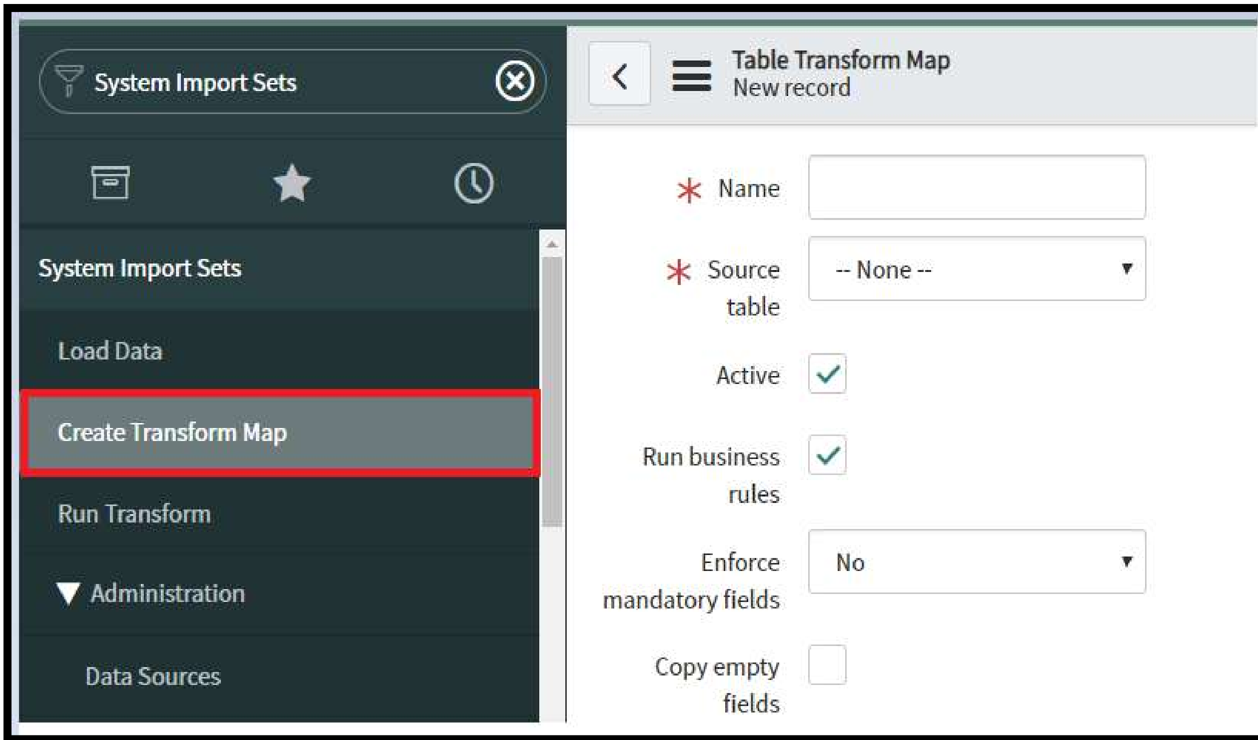
5. Fill required detail below



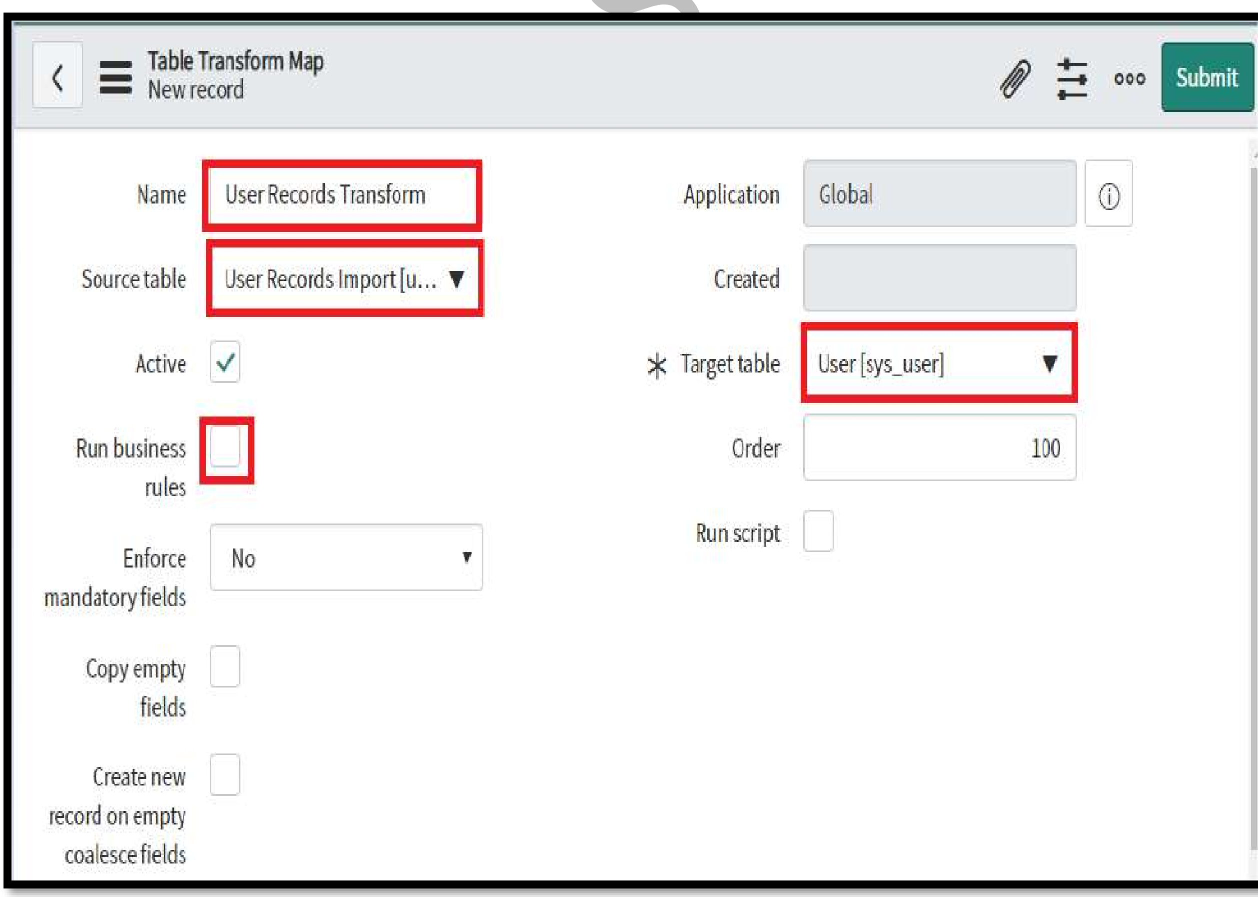
6. Select **Create Table** Radio Button
7. **Label:** User Records Import
8. **Name:** Auto Filled
9. **Source of the Import:** File
10. **File:** Select our target file from browser
11. **Sheet number:** 1
12. **Header row :**1
13. Click on **Submit**
14. Check the status of import process like below screen shot



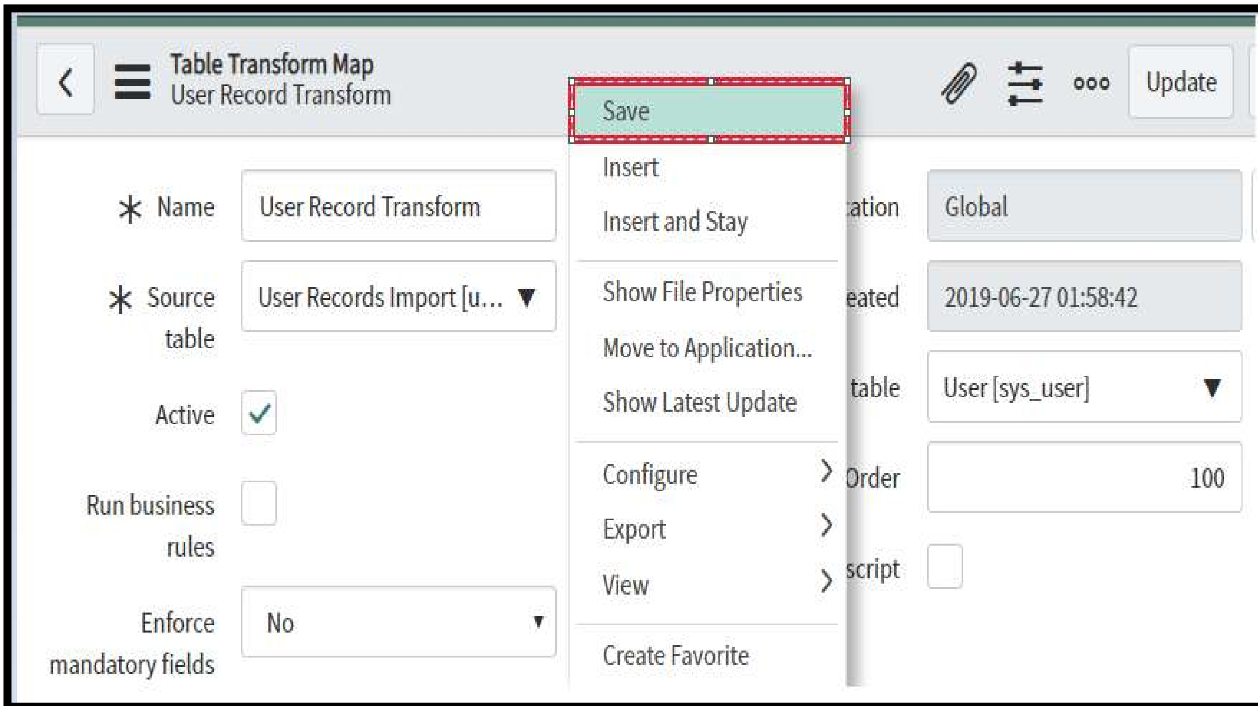
15. Create new **Transform Map**



16. Fill **Transform Map** form

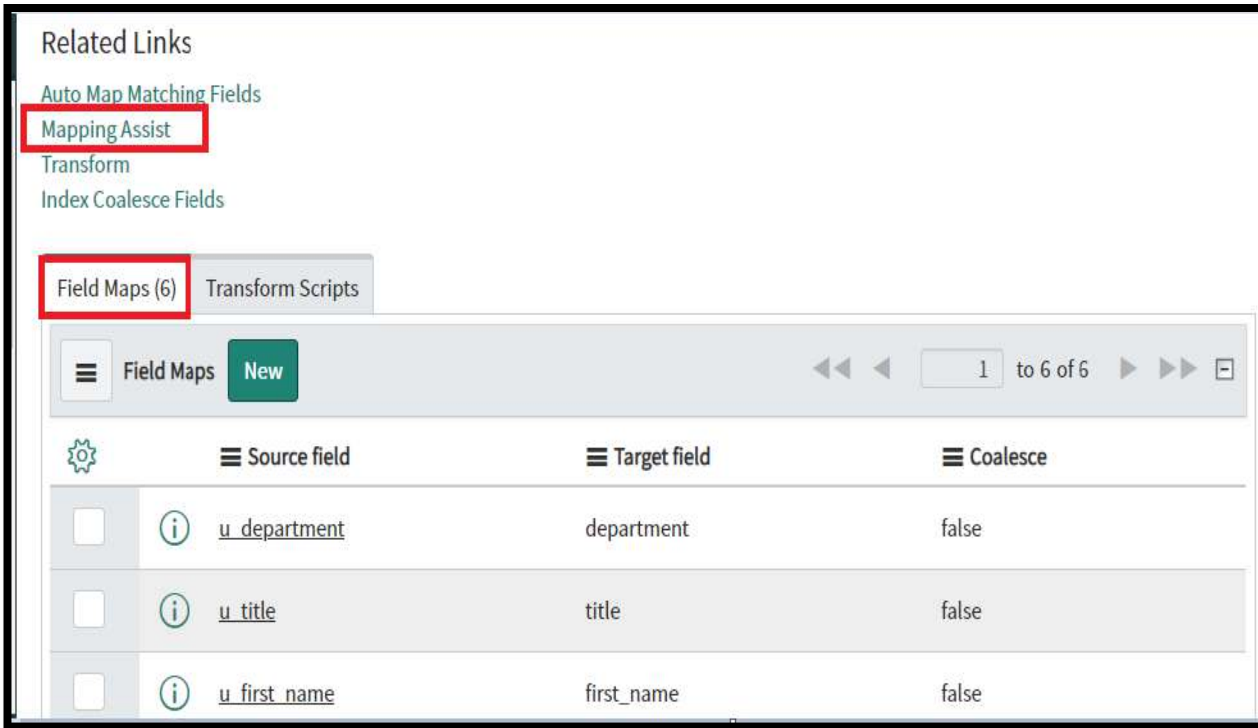


17. Name: User Records Transform
18. Source table: User records import
19. Target table: sys_user
20. Run business rules: Disabled property

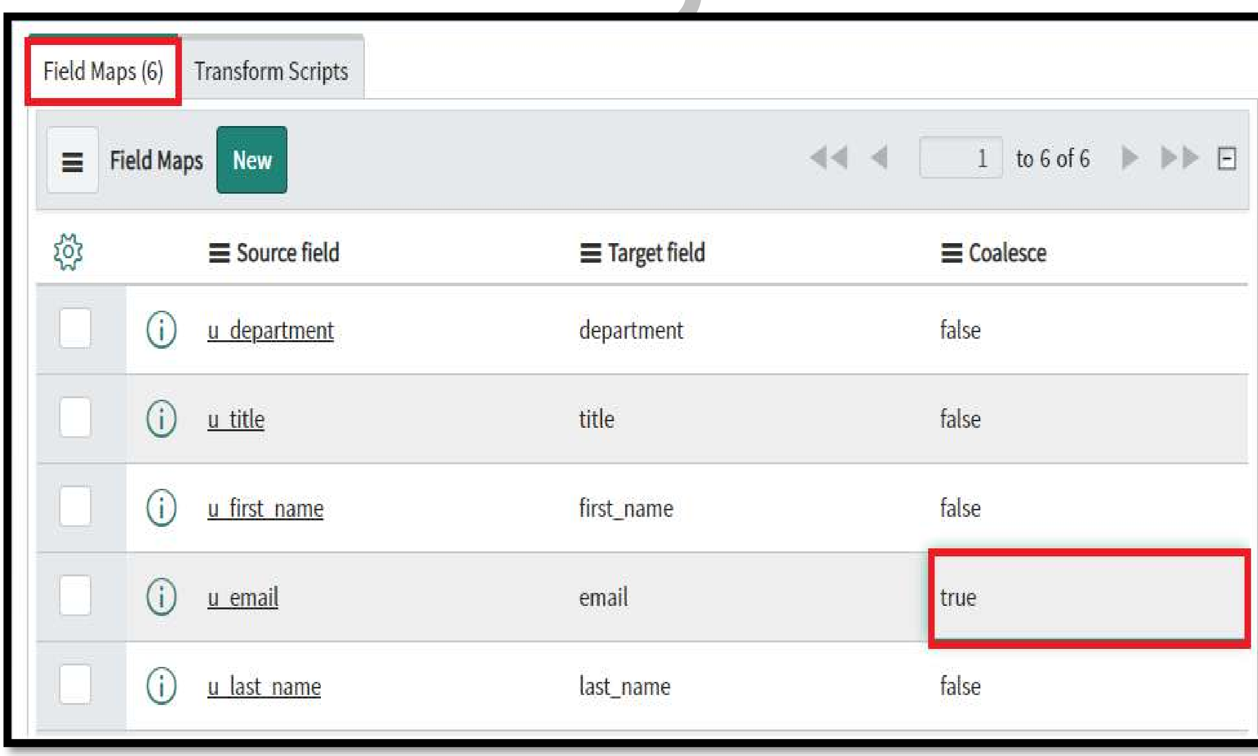


21. Right click form header and click on **Save**

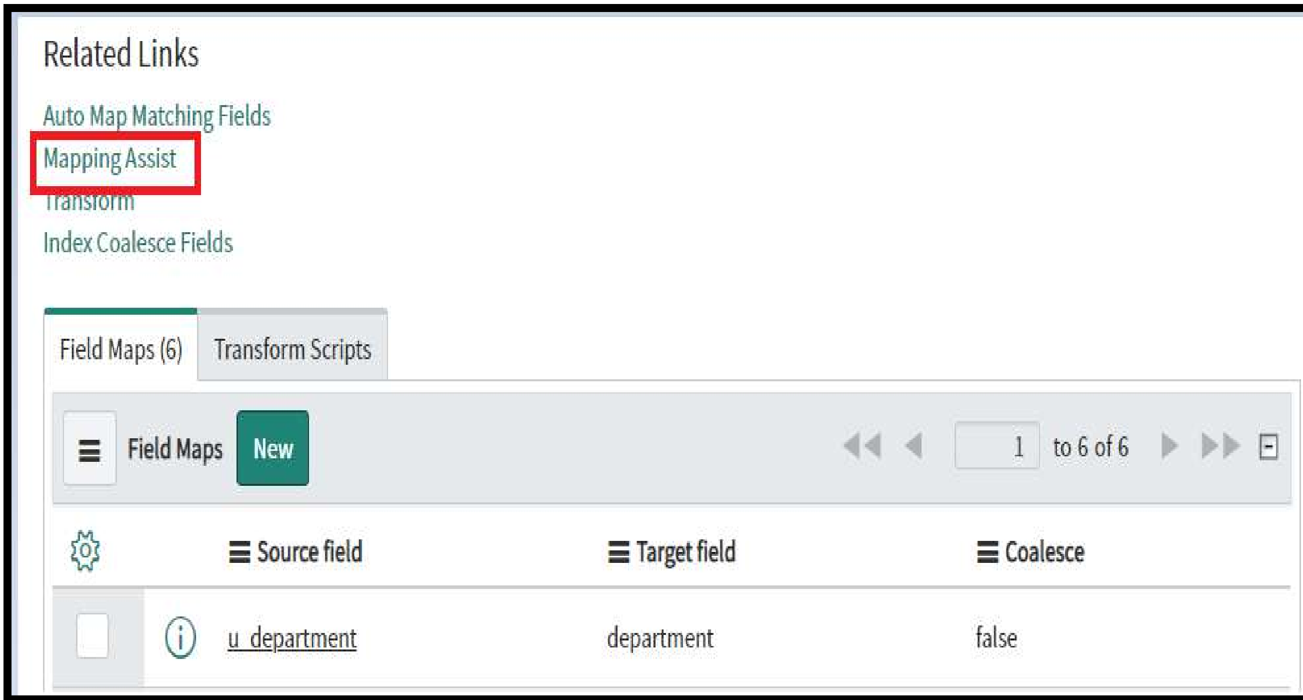
22. Down to under **Related Links**
23. Click on **Auto Map Matching Fields**

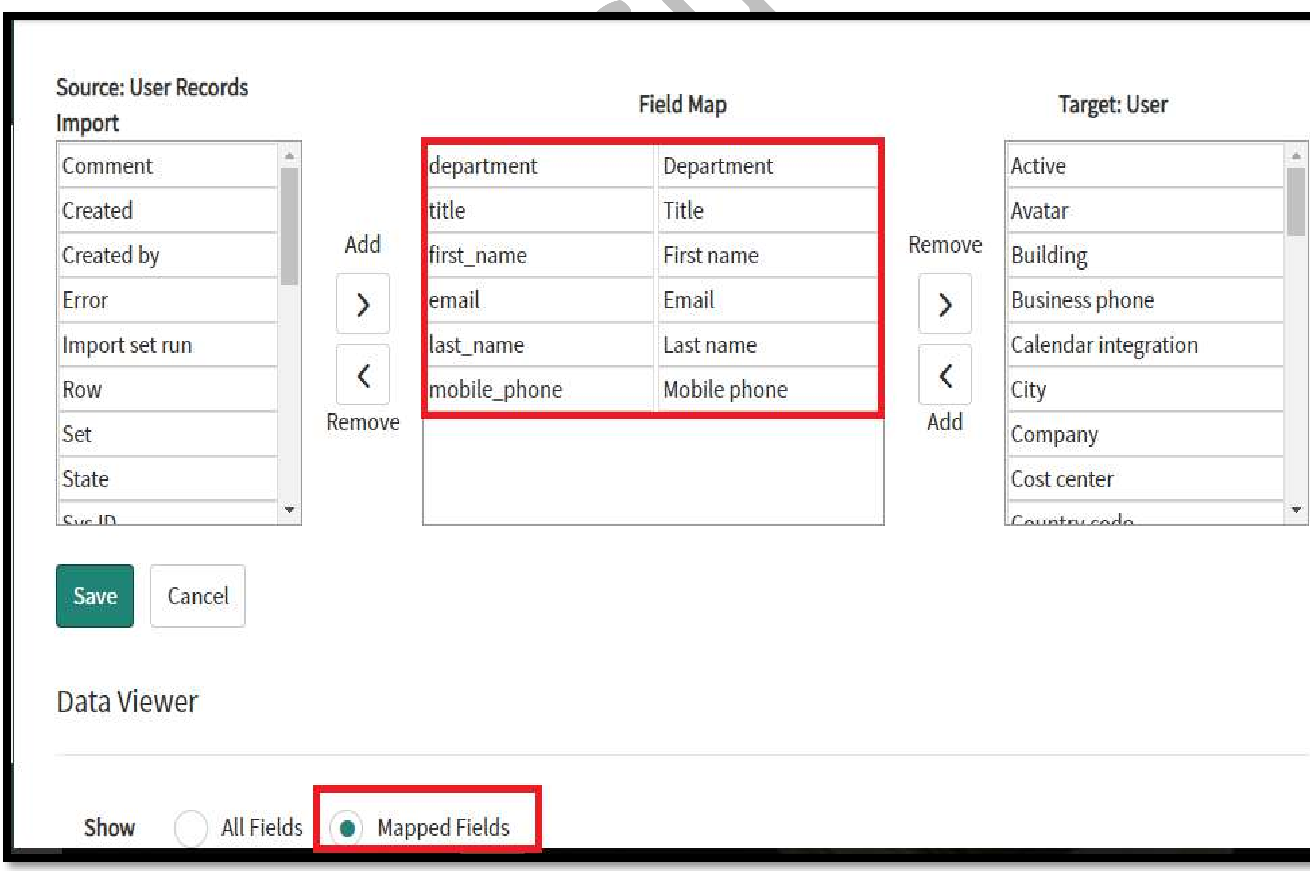


24. Create **Coalesce** for **email** field

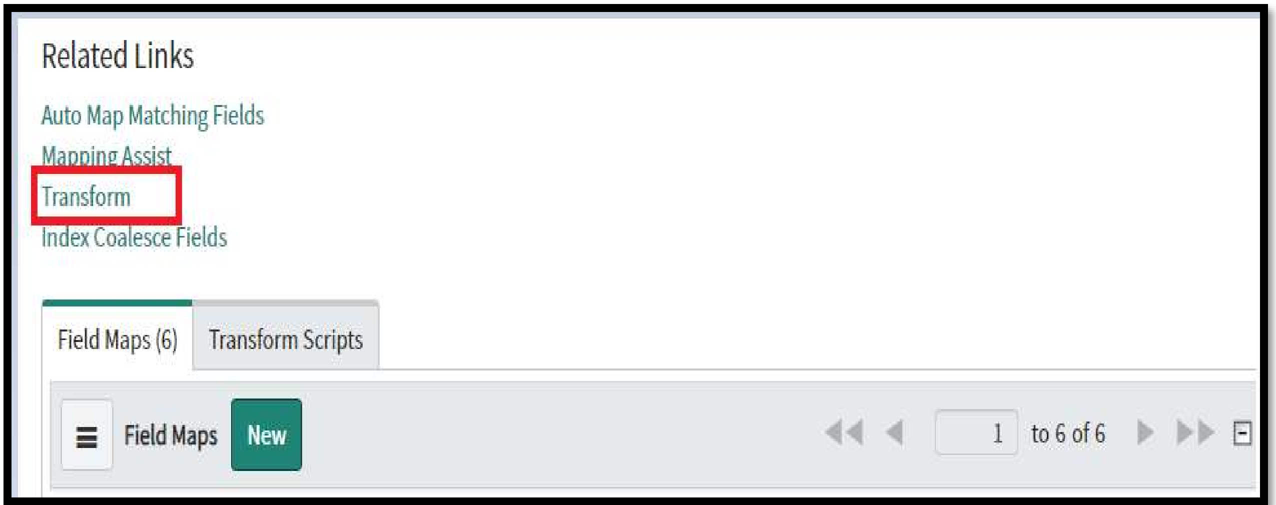


25. Click on **Mapping Assist**

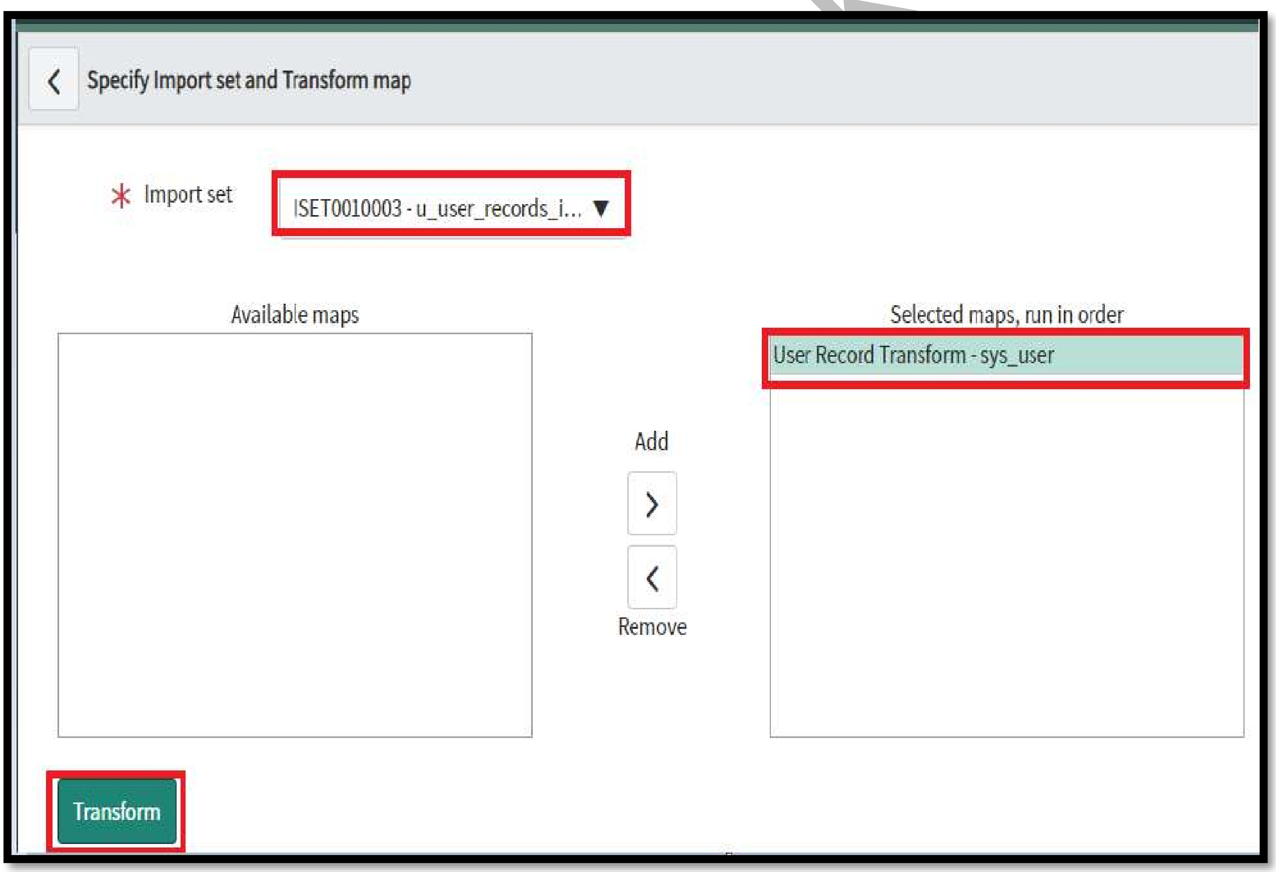




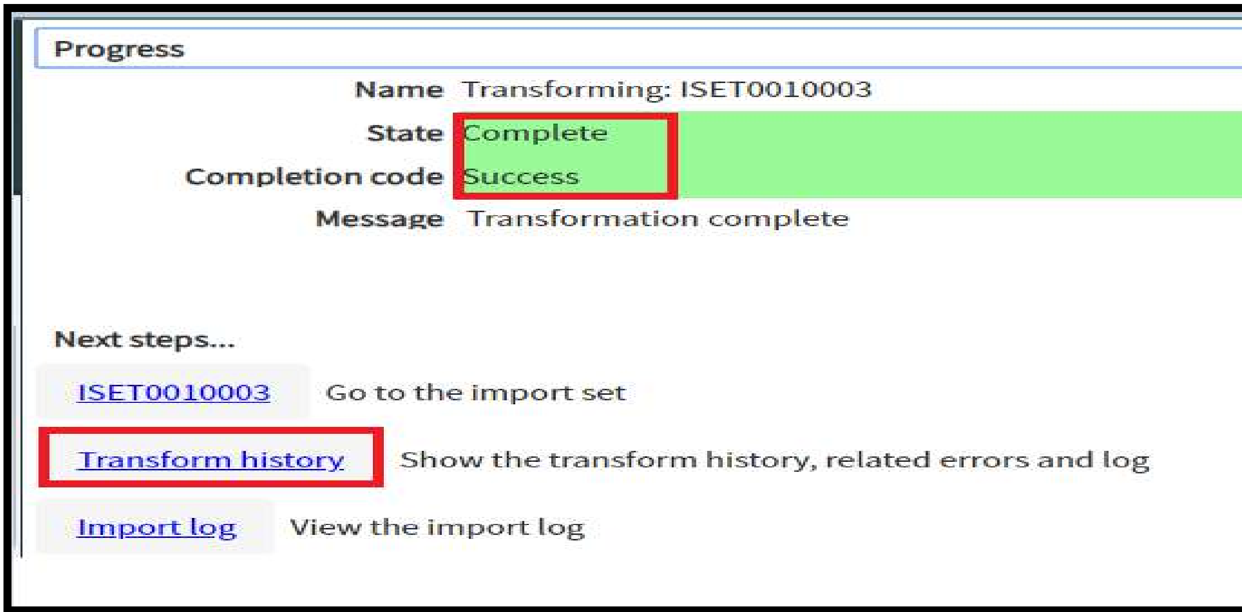
26. Select **Mapped Fields** radio button
27. Click on **Save** button



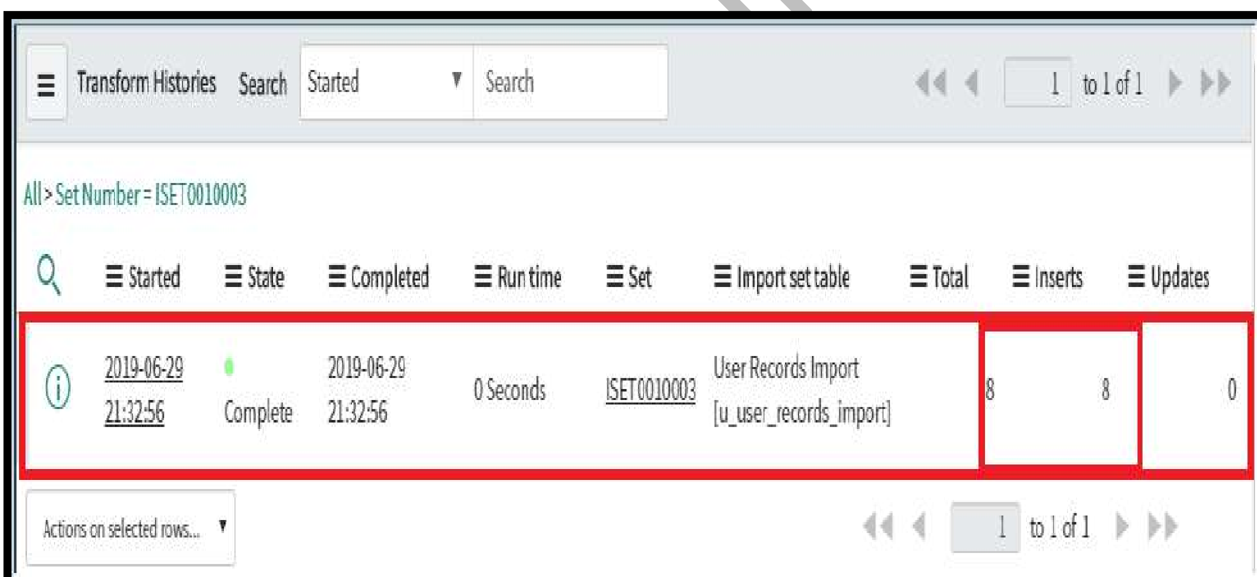
28. Click on **Transform**



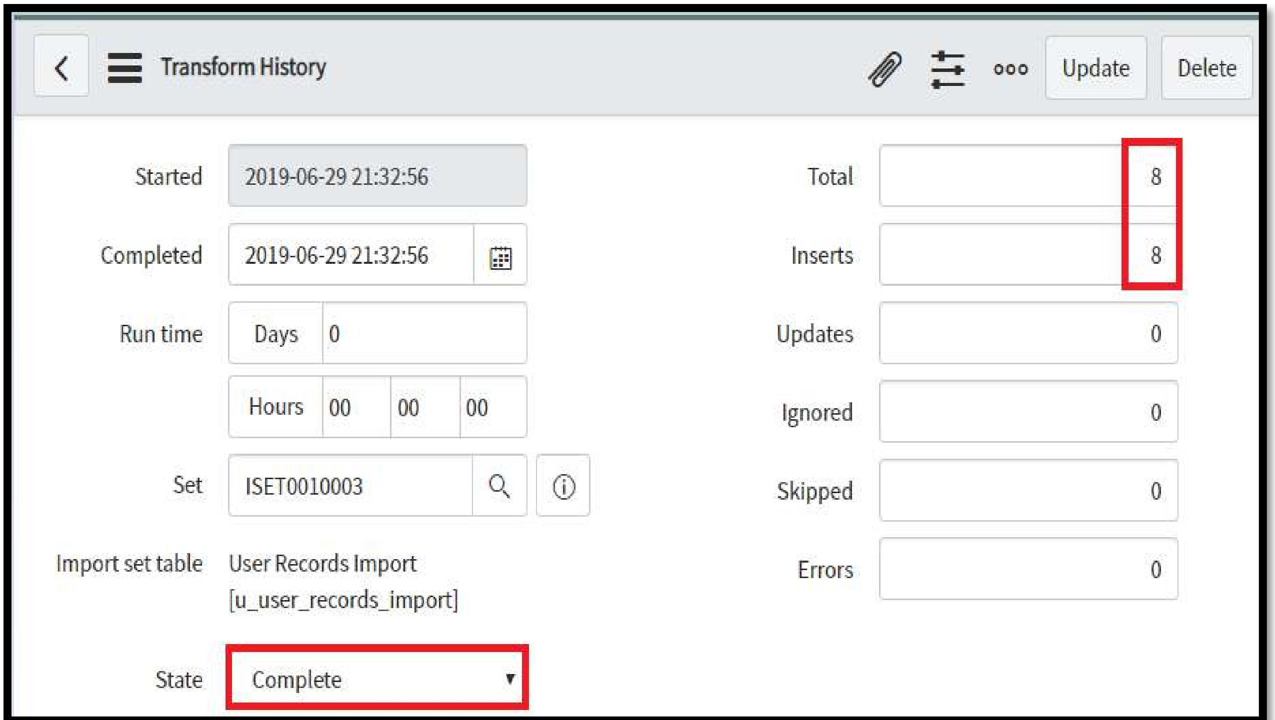
29. Select our **Transform**
30. Click on **Transform** Button



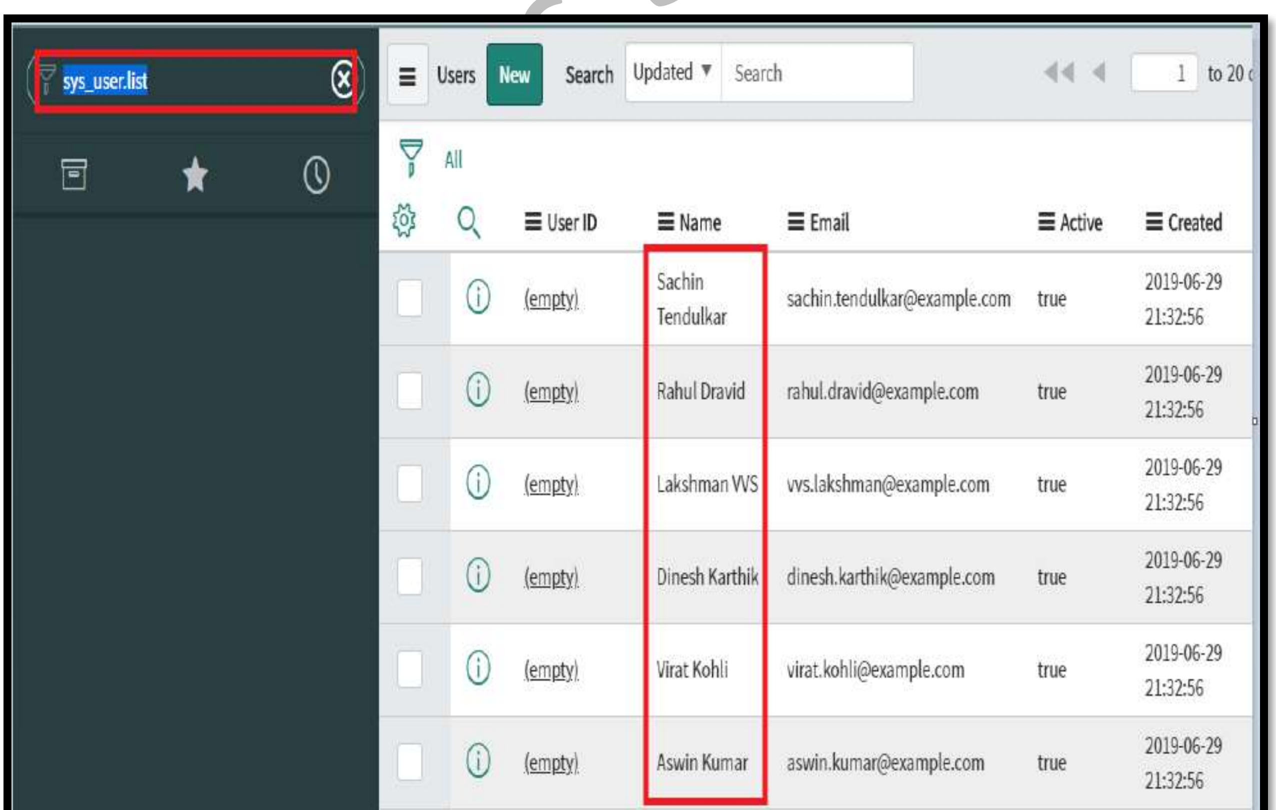
31. Click on **Transform History**



32. Check **Status** for all records
33. Open **ISET0010003** record check history



34. Navigate to **sys_user** table check all new records



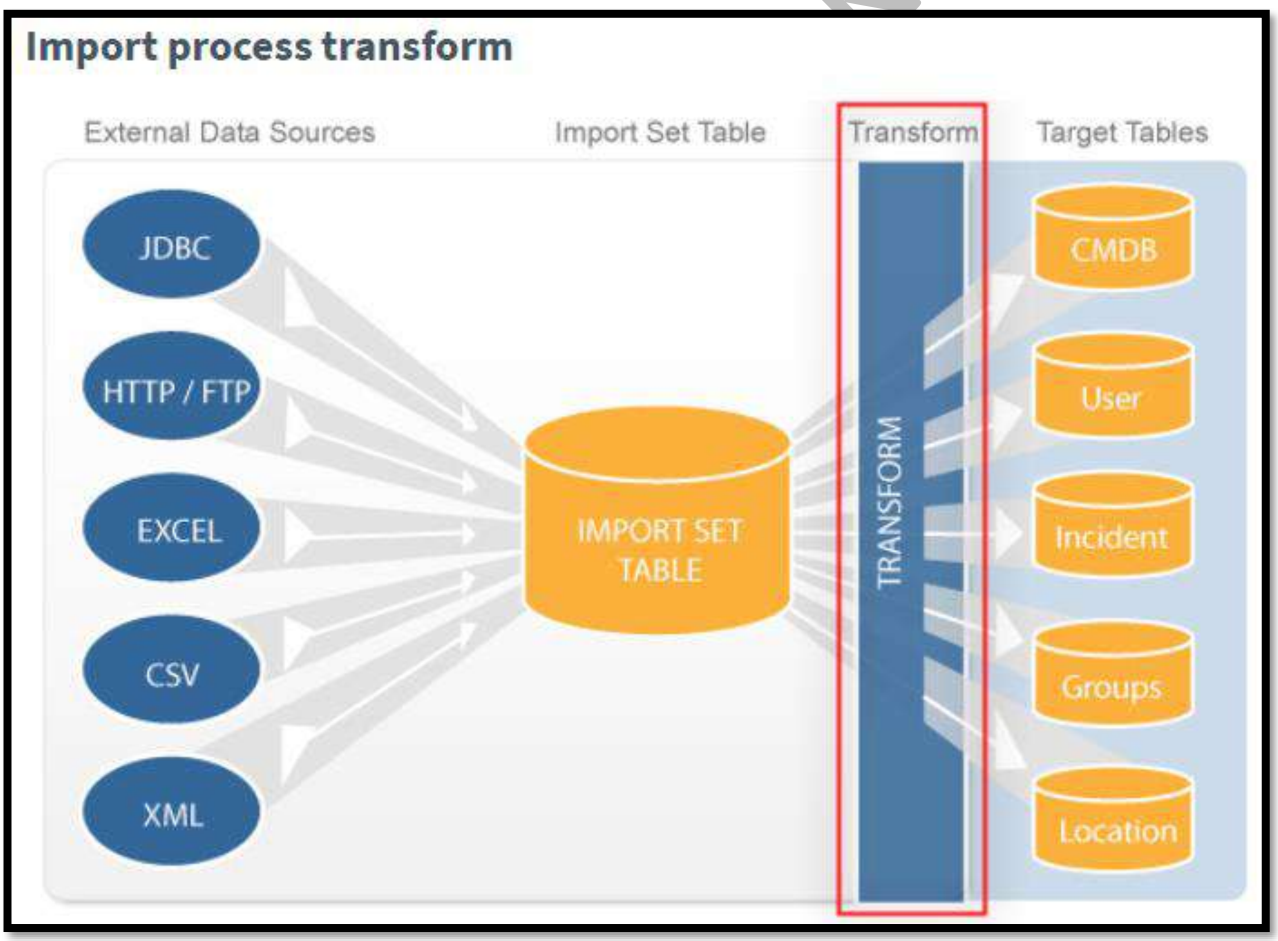
Import Sets Terminology

1. Load Data

If we want import bulk data into service now tables, first we have import the staging table (temporary table).

2. Transform Maps

A transform map is staging area that is used to set of field maps that define the relationships between fields in an import set and fields in an existing Service Now table **(Source & Target),** such as Configuration Item **[cmdb_ci]** or **User [sys_user].**



After created a transform map, we can be able to reuse it to map data from another import set to the same table based on our business requirement

The **Transform Maps** module enables an administrator to define destinations for imported data on any tables. Transform mapping can be as **simple as a drag and drop operation** to specify linking between source fields on an import set table and destination fields on any table. Use transform mapping to map source and destination fields dynamically.

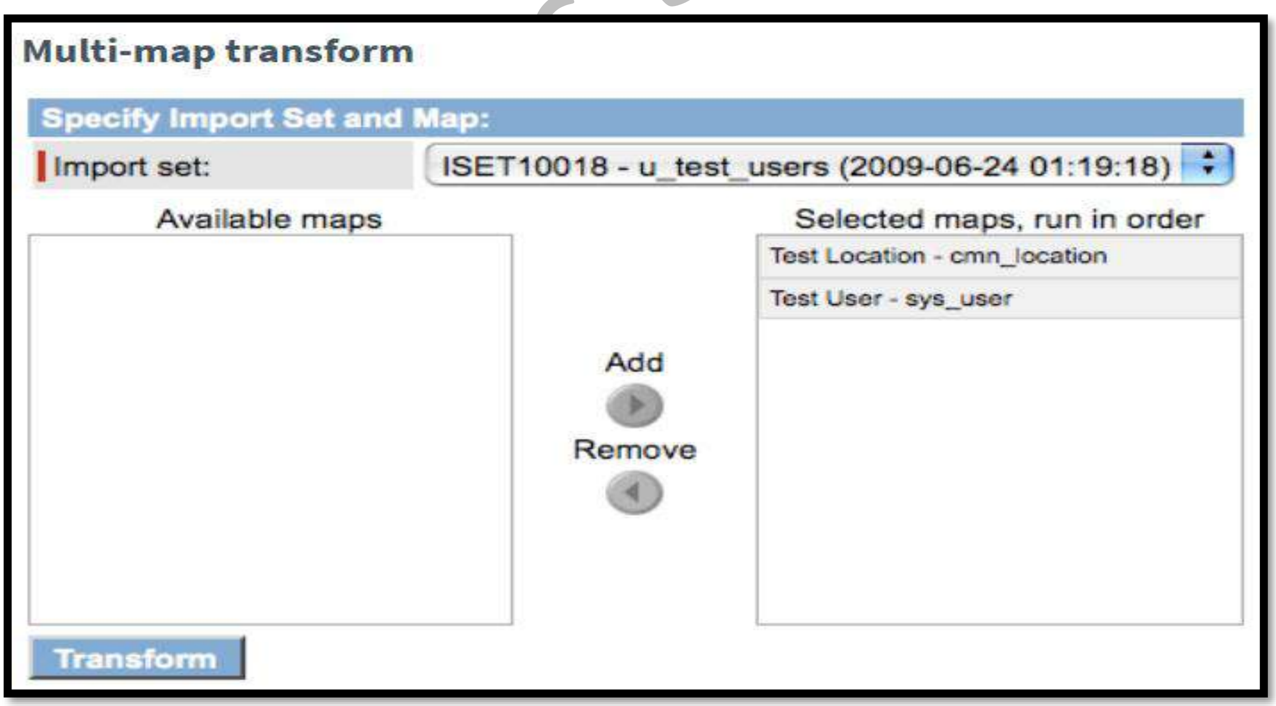
Multiple Transform Maps

Multiple transform maps can be applied to a single data source.

One import set row is created per transform map. This behavior can cause a large number of temporary records to be generated.

If we want use multiple transform maps for the same import set, the transform creates multiple entries in the import set table.

Users can select multiple transform maps during data import. The selected transform maps will be executed on the same import set in the order specified.



What is Coalesce

1. The coalesce is a field property that is used to when we inserting data into target table, System won't be accepting user to inserting duplicate records and and uses to updating existing records.

2. The coalesce option on a field map allows you to specify if the selected **Target field** should be used to coalesce on when import set records are transformed. If the field map **Coalesce** checkbox is selected,

3. when the import set row is transformed the instance checks for an existing record in the target table that has the same value in the **Target field** as the import set row **Source field**.

4. If an existing record with a matching value in the target table, then that record is updated.

5. If any record not matched, then a new record will create in the target table.

When we can choose fields in the target table to make coalesce on only if those fields will have unique values in the table. If more than one record in the target table matches the specified coalesce options, only the first matching record in the target table is updated.

No coalesce

If no coalesce is created, all imported rows or records are treated as new records in the target table. existing records are not being updated.

Single field Coalesce

You can coalesce on a single field to update an existing record.

If a target table record exists with the same value in the coalesce field as the staging table record, the target table record is updated using the import set record values.

Multiple-field coalesce

We can make **Coalesce** on multiple fields to update an existing record.

If a target table record exists with the same values in all coalesce fields as the staging table record, the target table record is updated using the staging table record values. All coalesce field values between the target and staging tables must match to coalesce with multiple fields.

Records Updating No Inserting (Restrict User)

When we want to only update the records where a record match is found, and skip where a record not match, then we can define a coalesce field and add the following script as an **onBefore** script to the transform map.

```
if (action == 'insert')
  ignore = true;
```

Working with conditional coalesce using dot-walking

We can use **dot-walked fields** in a conditional coalesce script, which is to match the email address of a user when importing **Incident or Problem** data.

In this example, this script is defined in the Source script of a field map for the Incident target table **sys_id** field.

```
var gr = new GlideRecord('incident');
gr.addQuery('caller_id.email', source.u_email);
//check if the incident caller's email matches the import row email value
gr.query();

if(gr.next())
{
    answer = gr.sys_id;
//if a match exists, return the sys_id of the matching Incident record
}
else
{
    answer= -1;
}
```

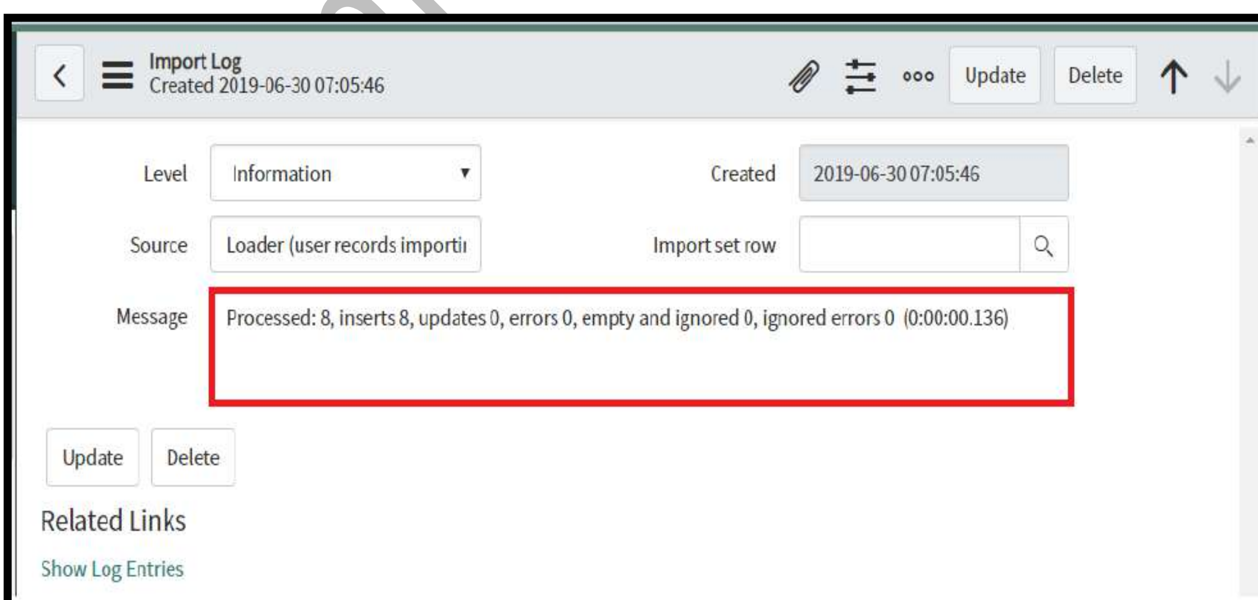
Working with Coalesce on empty fields

1. We can control user if an import set will coalesce on fields with no value.
2. The field map Coalesce empty fields field allows you to coalesce on fields with no value.
3. By default, fields marked as Coalesce in the field map cause the import set to check for a target record with a field value that matches the value from the import set staging record.
4. When Coalesce empty fields is selected for that field map, an empty value in the target record and import set staging record counts as a match for the purpose of coalescing.
5. the User transform map coalesces on the email field. With the Coalesce empty fields option selected,
6. a source record containing an empty email address coalesces to a target record containing an empty email address.

Import sets maximum row size

1. Rows imported using import sets must not exceed the maximum row size.
2. A single row in a database may not contain more than 8126 bytes of data.
3. The size of each row is determined by the amount of content in all fields, as well as the character set for text fields.
4. For example, a row with 10 text fields each containing 1000 characters using a French character set takes 15360 bytes.
5. Attempting to import more data to a single row than the maximum size causes the import to skip that row. Any rows that were skipped for this reason are listed in the import log.

View the Import Log File



Transform map scripts

Transform Map scripts allow user to customizing and performing import operations using an automatic programming interface to including advanced logic.

A transform map script will execute as events fired while an import set is being transformed onto a Service Now table. Transform Map scripting is fully integrated into the Service Now scripting environment.

There are two types of Transform Map scripts:

1. Explicit Transform Map scripts, which explicitly define mapping relationships
2. Transformation Event scripts, which modify the processing of events at different stages of a transformation

Types of Transform Map Script Events

1. **onStart**
2. **onComplete**
3. **onBefore**
4. **onAfter**
5. **onForeignInsert**
6. **onChoiceCreate**
7. **onReject**

Transform Map Script Events

1. onStart

The **onStart** event script will execute at the start of an import run, before any data rows are read.

```
var info = "Starting Import Set Transformation Mapping operation using Data source, "
+ map.table_source + " and Transform Map " + map.name + " on import set " + import_set.number;
log.info( info );
```

2. onComplete

The **onBefore** event script will be execute at the start of a row transformation, before the source row is transformed into the target row

```
// Create a myimport_completed event that can be reacted by an email notification or script action
// (there is already an import.finished event that the system will create at the end of an import)
var e = new GlideEvent("myimport_completed", import_set.sys_id, map.sys_id, "");
e.insert();
```

3. onBefore

The **onBefore** event script will be execute at the start of a row transformation, before the source row is transformed into the target row.

```
var is = source.u_name.toString();
var info = "Before the row is transformed, " + is;
log.info( info );

// Make sure a company name has been provided
var company = source.u_company.toString();
if(company.length == 0 ){
  ignore = true;
  info ="No company name, row ignored! " + is;
  log.info( info );
}
```

4. onAfter

The **onAfter** event script will be execute at the end of a row transformation, after the source row has been transformed into the target row and saved.

```
if(source.new=="true"){
  gs.include('Cart');
  var bundle =new GlideRecord('sc_cat_item');
  bundle.addQuery('name','CONTAINS','comp');
  bundle.query();
  if(bundle.next()){
     var new_comp_cart = new Cart();
     var cart = new Cart();
     var item = cart.addItem(bundle);
     var rc = cart.placeOrder();
     log.info(rc.number);}}
```

5. onForeignInsert

The **onForeignInsert** event script will be execute at the start of the creation of a related, referenced record, before the record is created.

```
//Create an event.
var e = new GlideEvent("myimport_ForeignInsert", action, name, "");
e.insert();
```

6. onChoiceCreate

The **onChoiceCreate** event script is processed at the start of a choice value creation, before the new choice value is created.

```
//Create an event
var e = new GlideEvent("myimport_ChoiceCreate", action, value, "");
e.insert();
```

7. onReject

The **onReject** event script will be execute during the occurrence of a foreign record or choice creation, and the foreign record or choice is rejected, the entire transformation row is not saved.

```
//Create an event
var e = new GlideEvent("myimport_ChoiceCreate", action, value, "");
e.insert();
```

Excericse:2 Schedule a data import

We can schedule imports make it is possible to specify that certain import activity should occur at some regular interval.

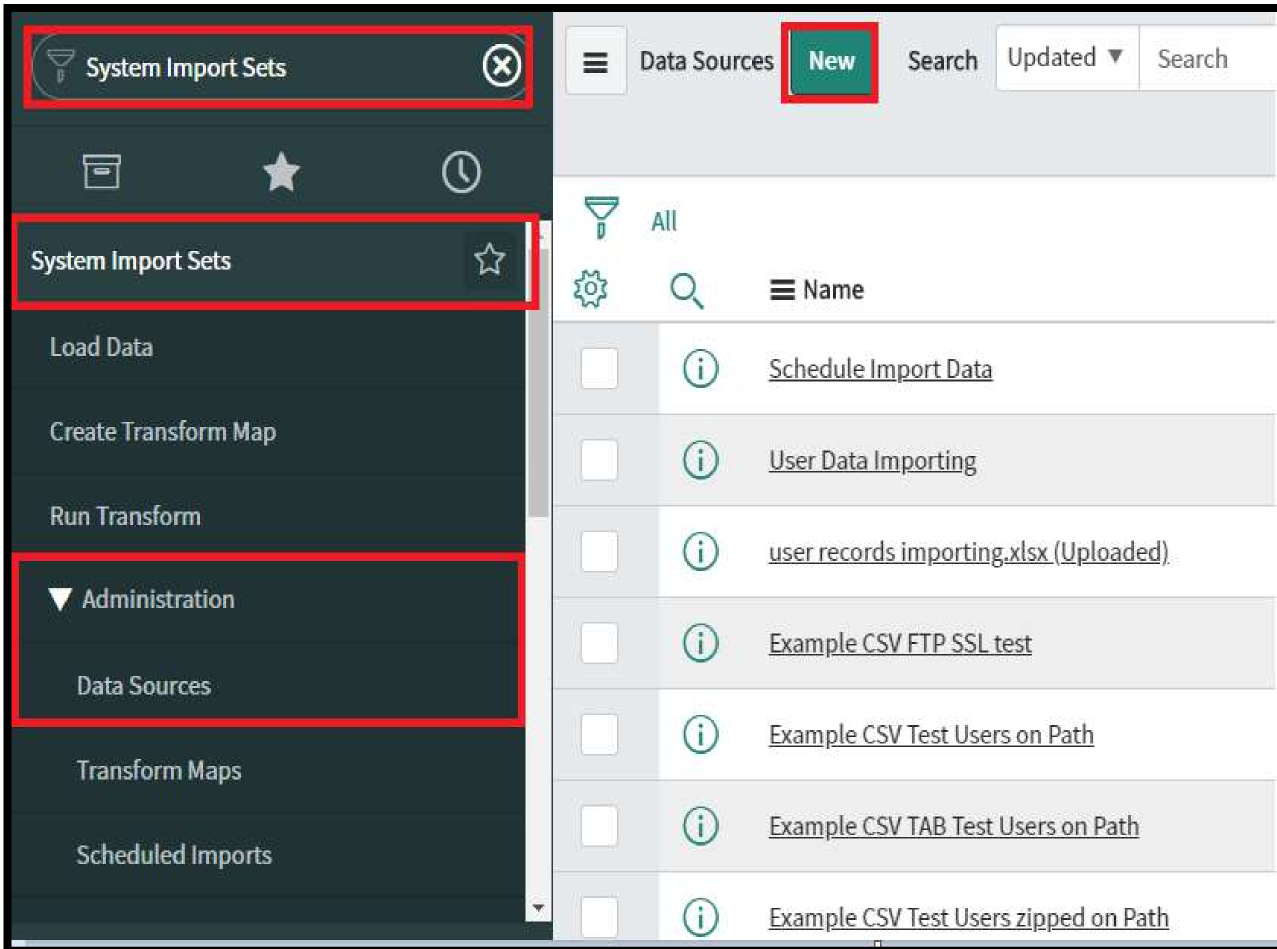
Do not schedule multiple imports at the same time. Running multiple imports concurrently may negatively impact performance or cause an instance outage.

Creating Data Source

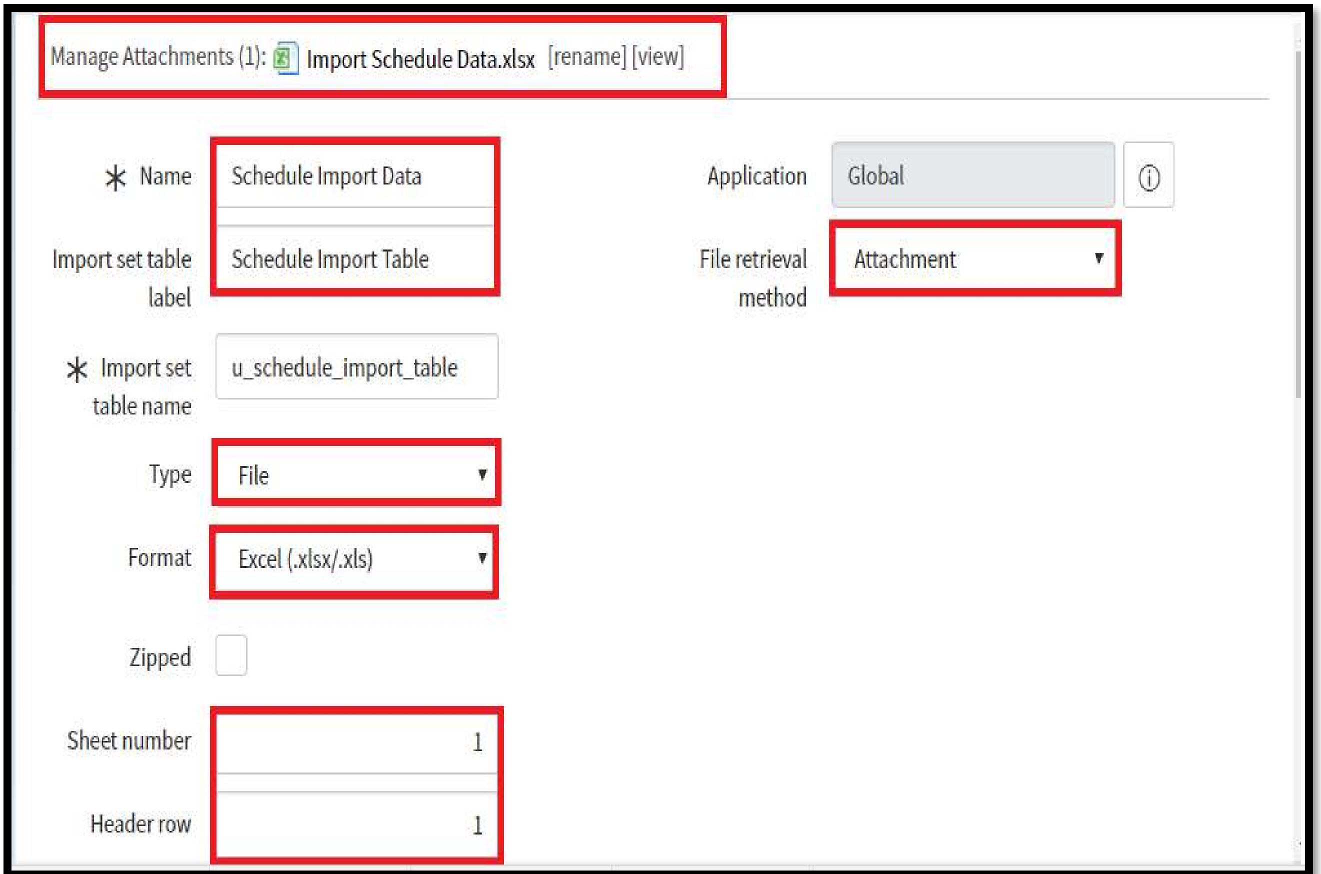
We can create a data source record to describe what data an import set should import.

Procedure

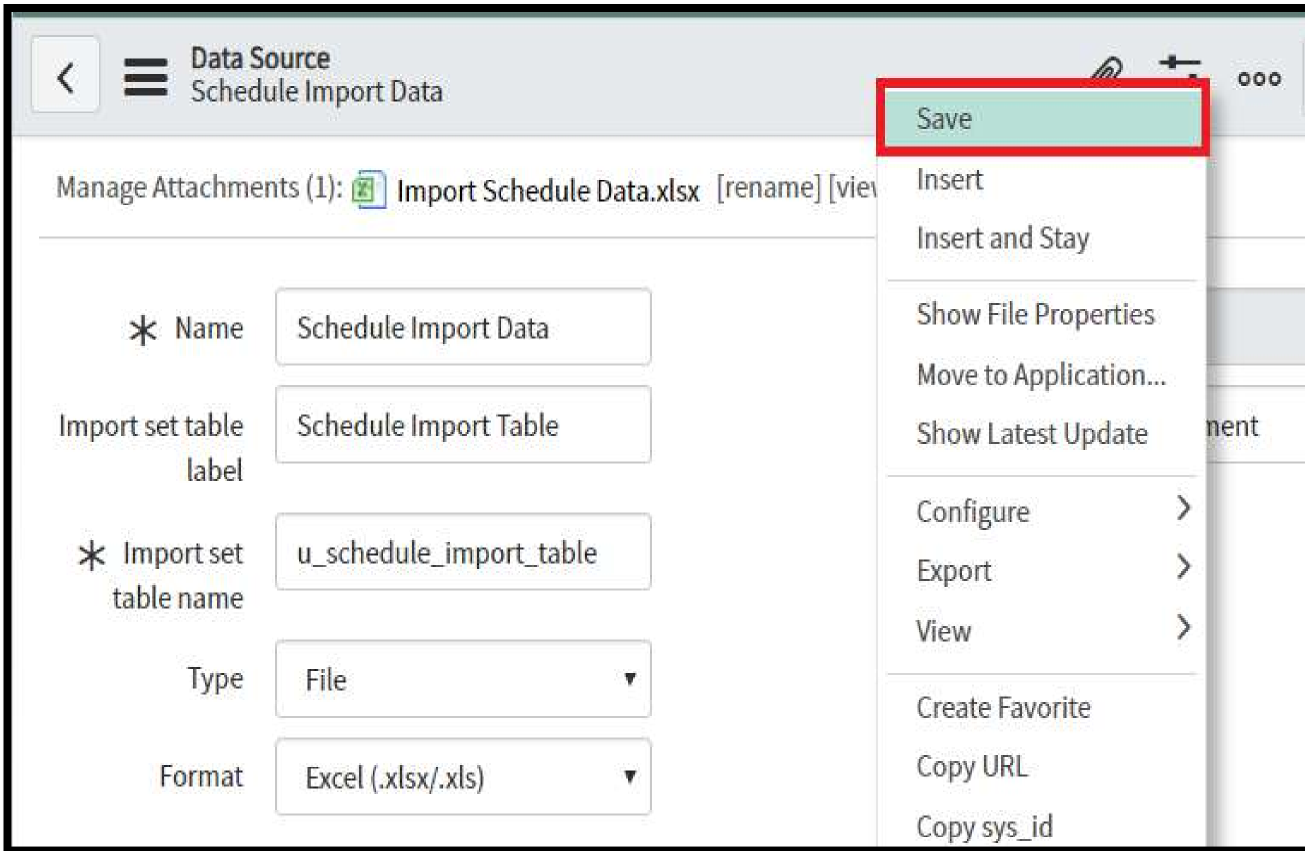
1. Navigate to **System Import Sets > Administration > Data Sources.**
2. Click **New**.



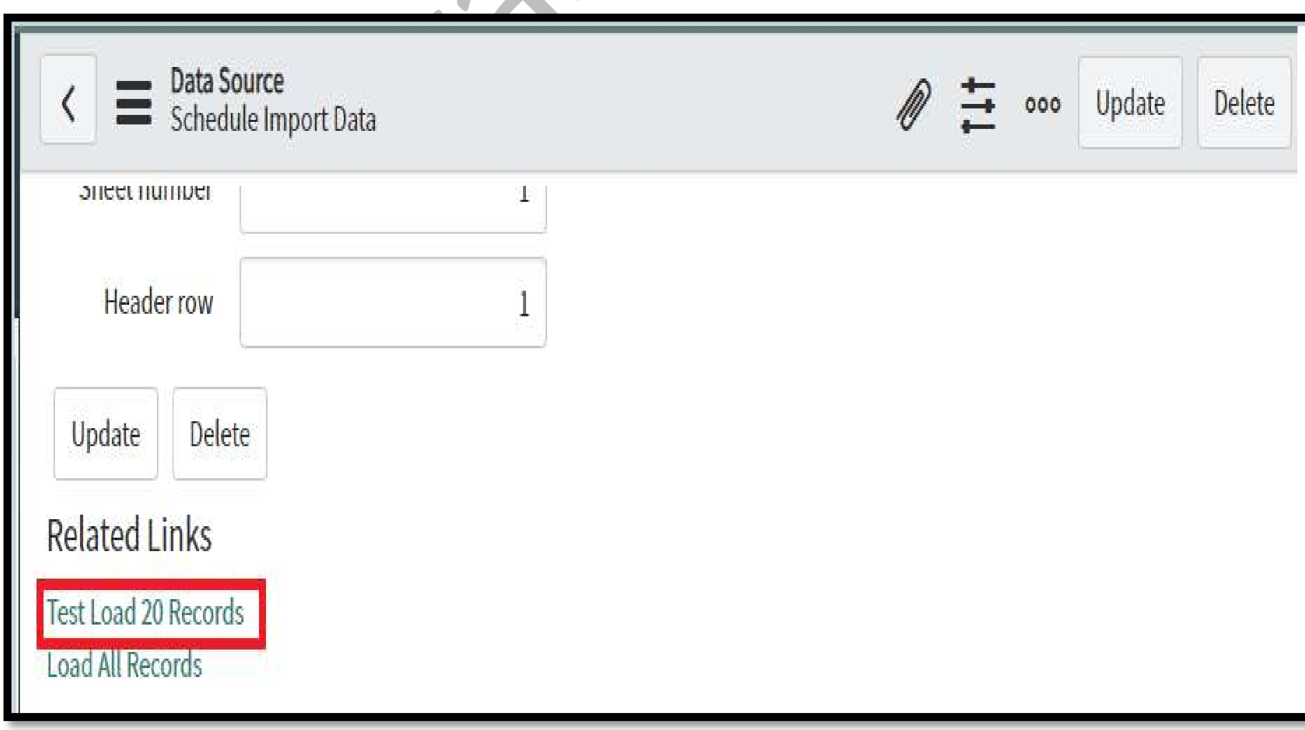
3. Fill **Data Sources** form



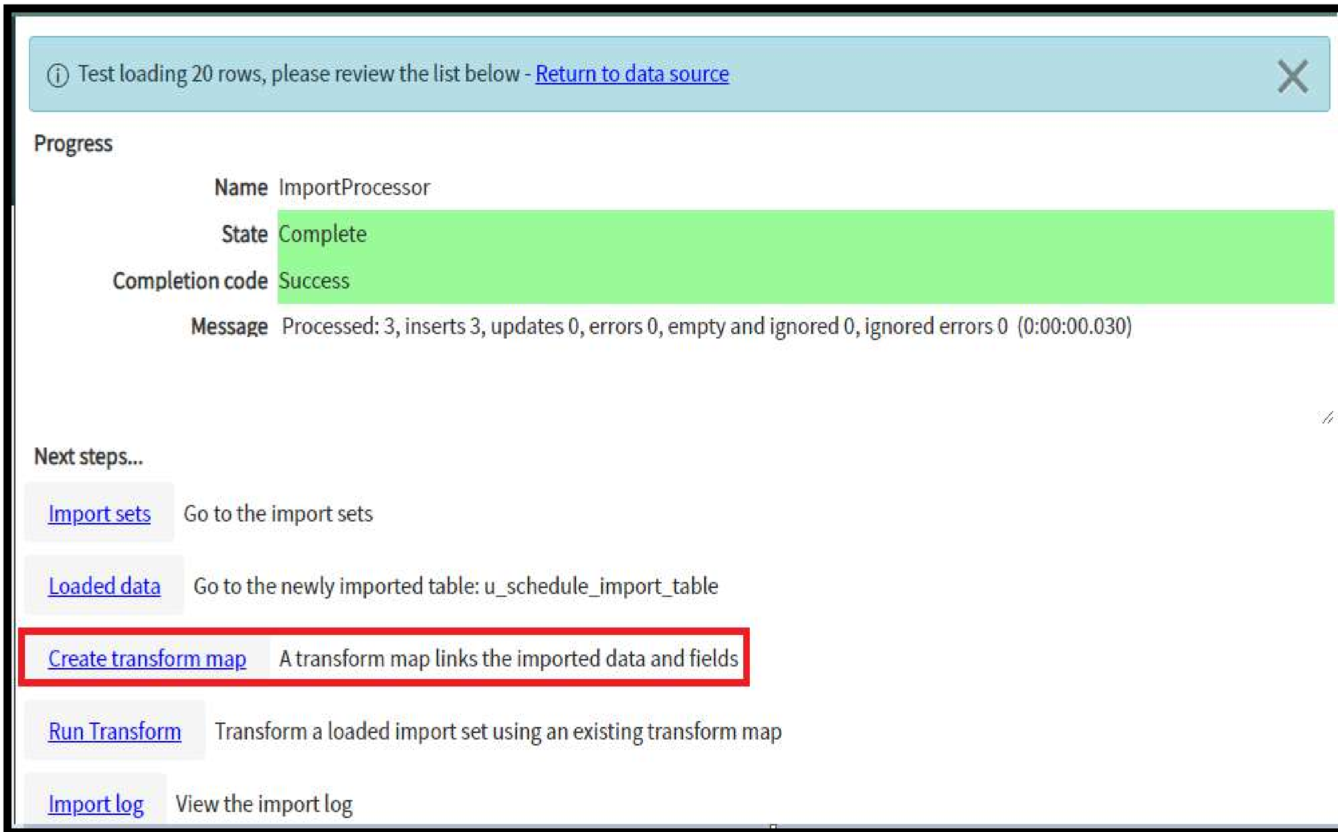
4. **Name:** Schedule Import Data
5. **Import set table label:** Schedule Import Table
6. **Import set table Name:** Auto Filled
7. **Type:** File
8. **Format:** Excel(.xls)
9. **Sheet Number:**1
10. **Header row :**1
11. **File Retrivievel method:** Attachment
12. **Manage Attachment:** Import Schedule data attached here



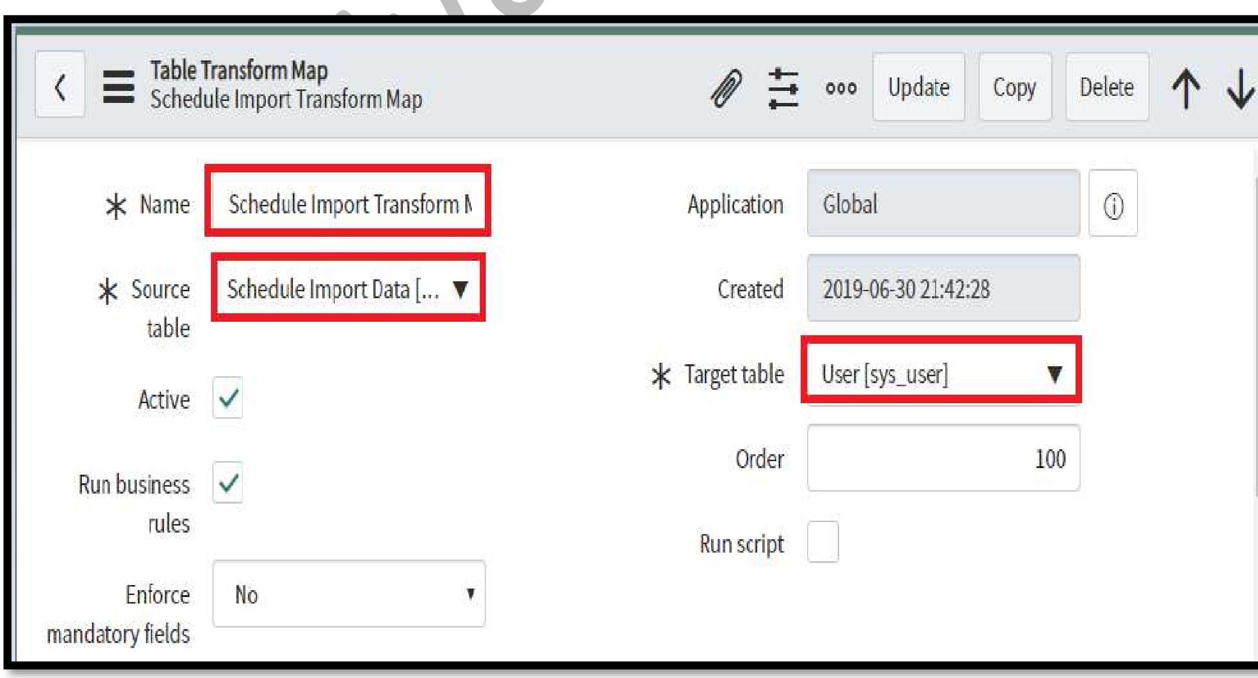
13. Click on **Test Load 20 Records**



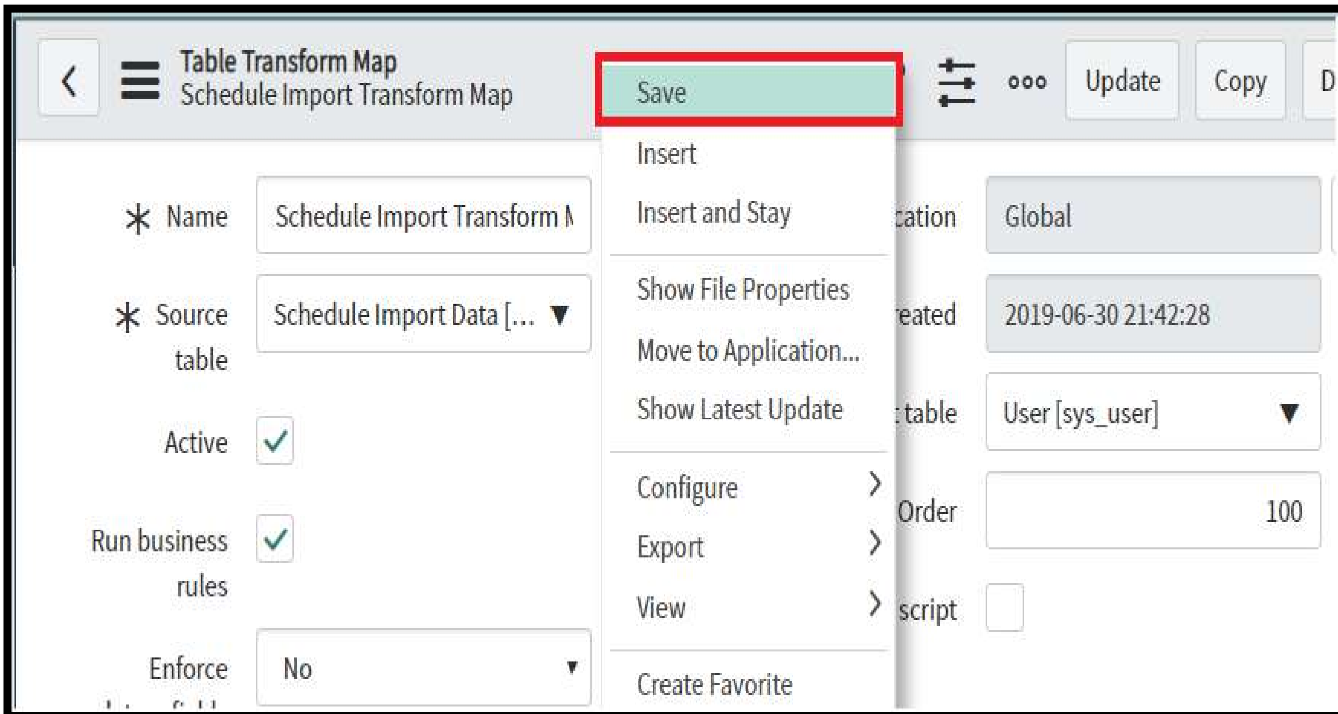
14. Create new transform map to load data into service now table



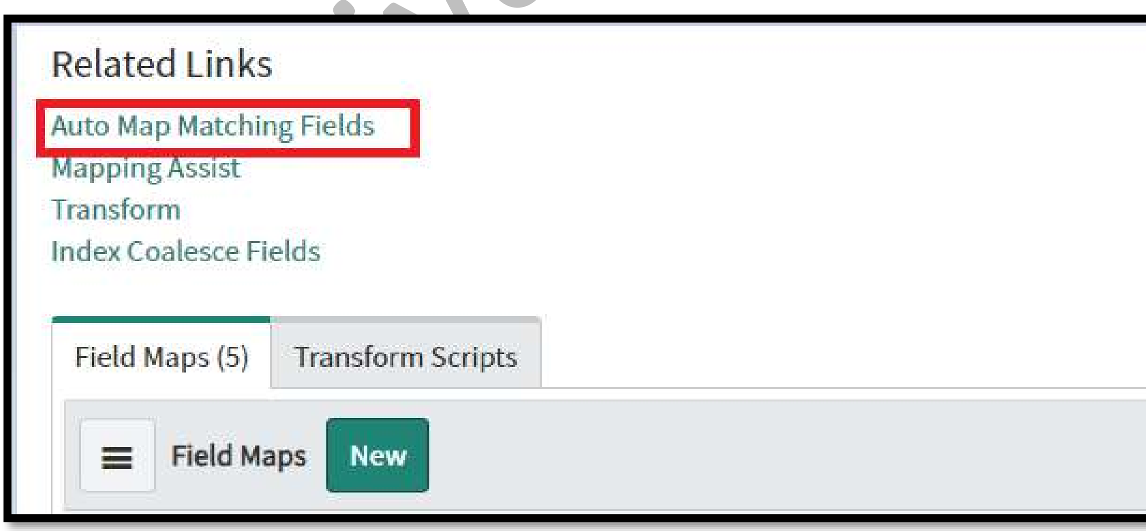
15. Fill transform map form



16. Name: Schedule Import Transform Map
17. Source Table: Schedule Import Data
18. Target Table:sys_user

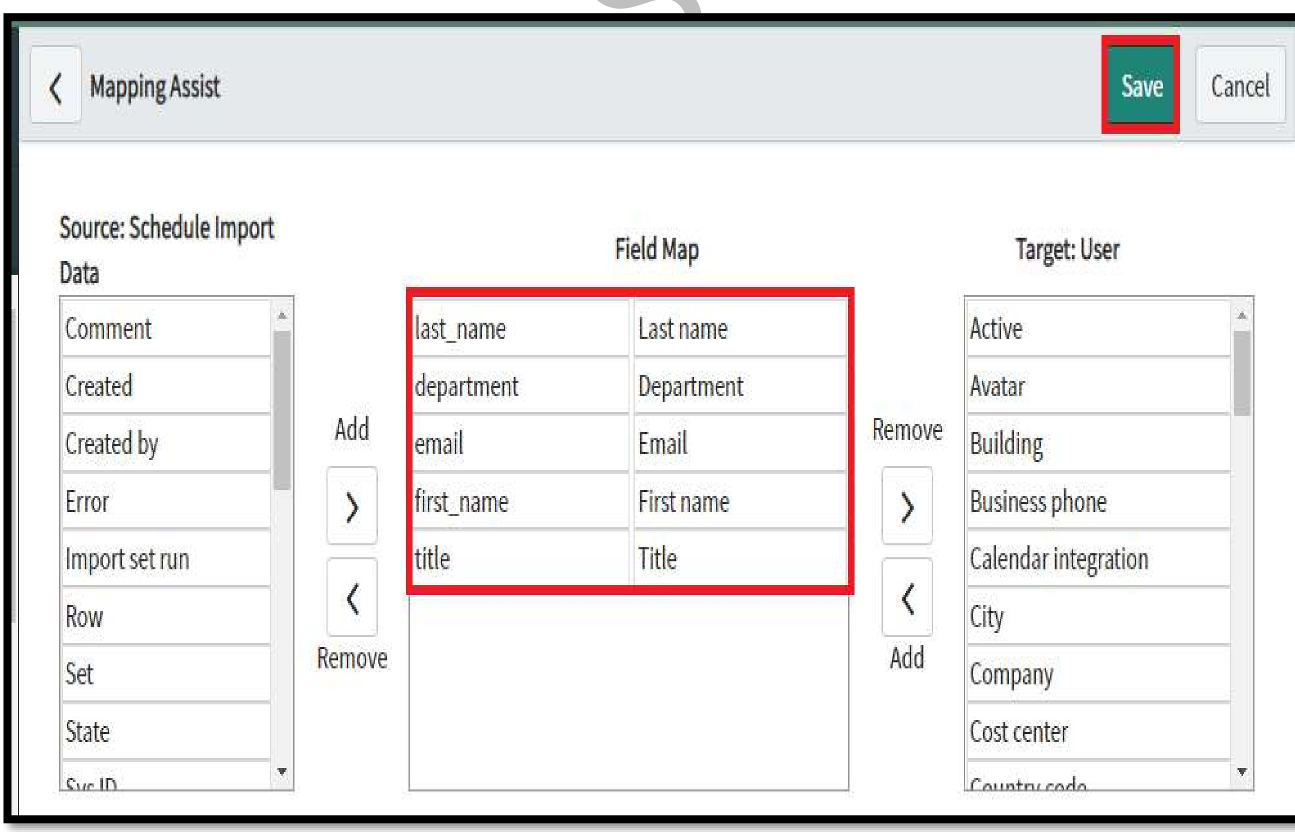


19. Right click on form header and click on **Save**
20. Down the same form click on **Auto Map Matching Field**



21. Click on **Mapping Assist**

22. Auto mapping fields from **Source** and **Target**



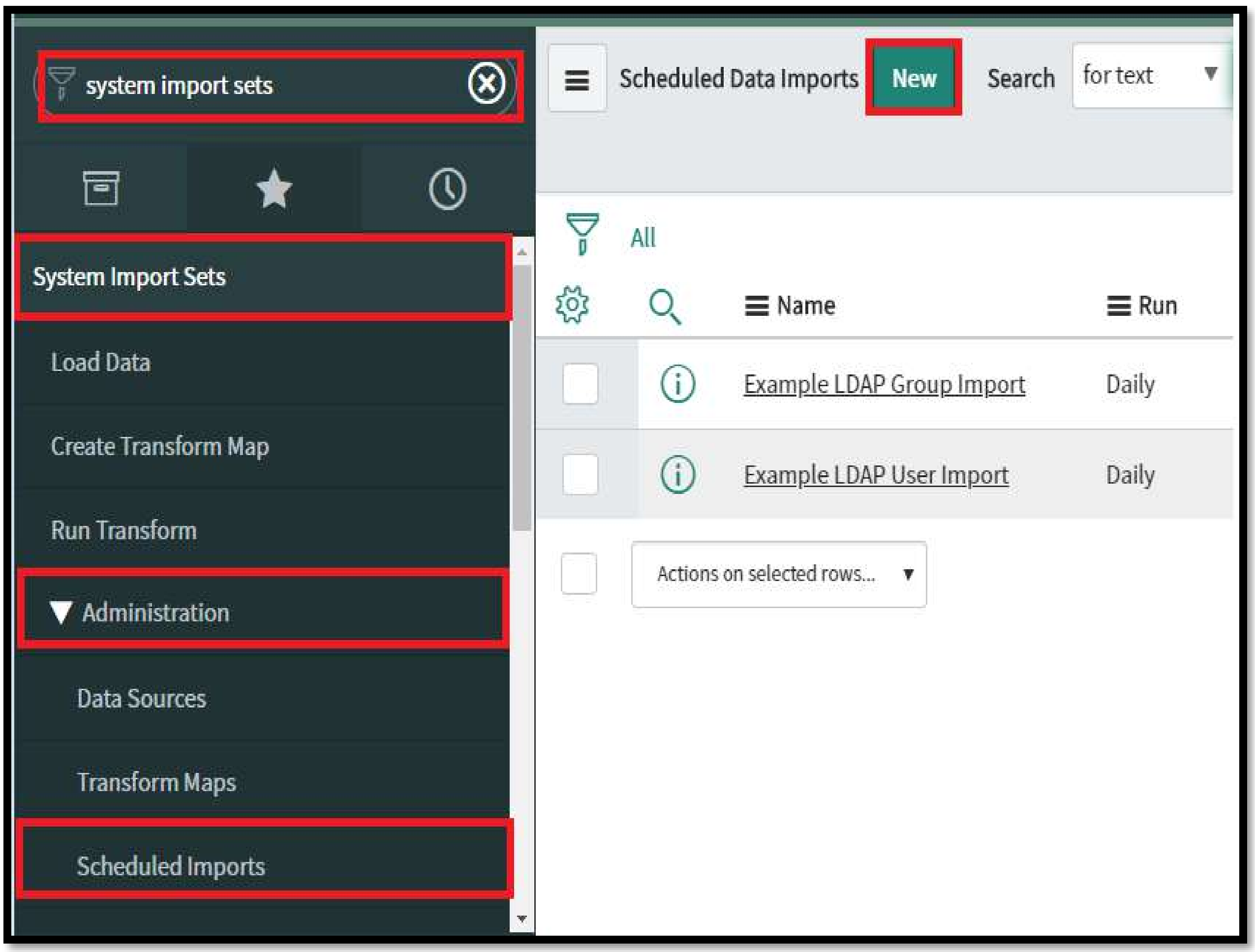
23. Click on **Save**

Schedule a data import

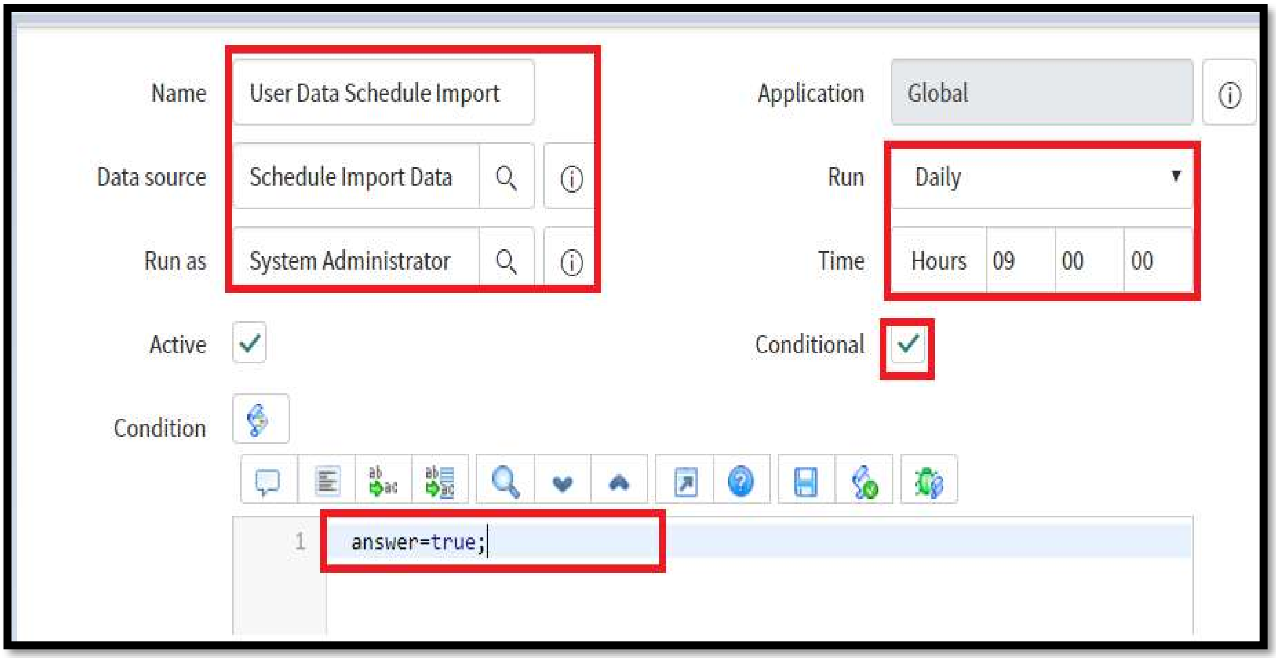
After created a data source we can Scheduled imports make it is possible to specify that certain import operations based on client requirement

Procedure

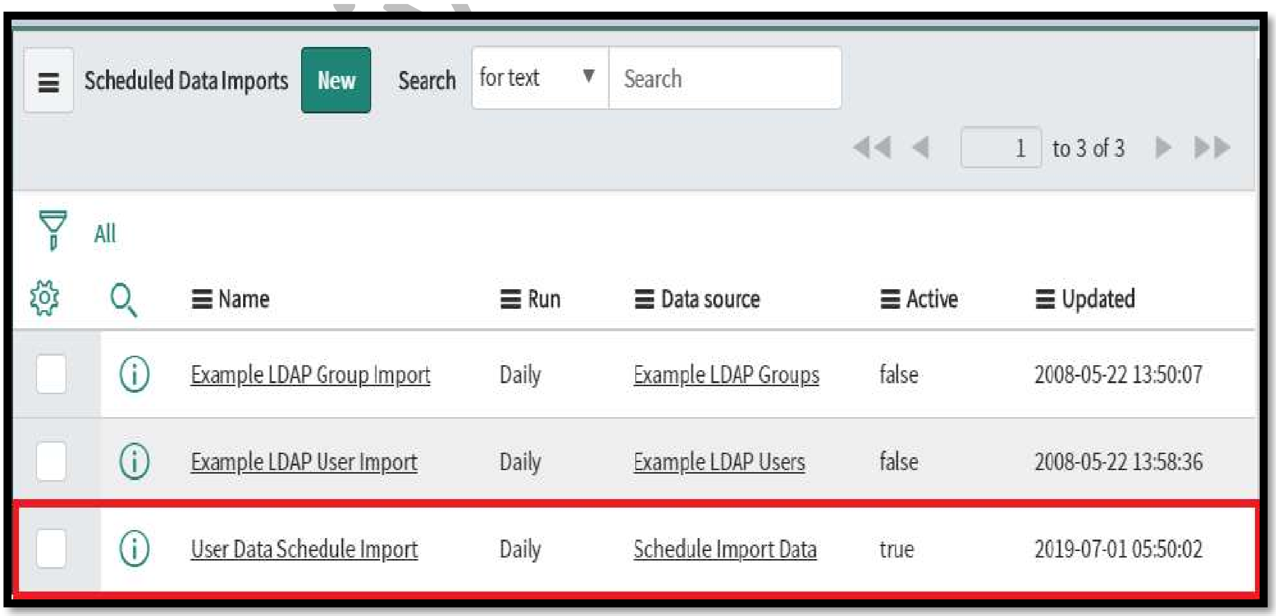
1. Navigate to **System Import Sets > Administration > Scheduled Imports**.
2. Click on Scheduled Data Imports **New**



3. Fill Data Import **Form**



4. **Name:** User Data Schedule Import
5. **Data Source:** Schedule Import Data **(Add Which you have created before)**
6. **Run as:** auto filled **(who logged in system already)**
7. **Run:** Daily
8. **Time:** 09:00:00
9. **Conditional:** Checked True
10. **Condition:** answer = true;
11. Click on **Submit**



12. Selected record will execute daily at 9 am

Chapter – 14

Update Sets

What is Update Set

An **Update Set** is a group of configuration changes or customizations that can be used to track all individual developer changes and moving changes from one instance to another. This feature will allow administrators to group a series of changes into a particular set and then move them as a unit to other systems for **Testing or Deployment** purpose.

Update sets will track all our developer changes to applications and system platform features. This allows developers to create new functionality on a non-production instance and moving the changes to another instance.

Note: An update set is an XML file that contains:

1. Each set of record details that should be uniquely identify the update set.
2. Group of customization or Configuration changes in our instance.
3. A state that can be defining whether another instance can retrieve and apply configuration changes.

Why use update sets

The **sys_id,** the unique identifier that marks every Service Now record would be different. This can cause issues in certain scripting and configuration situations.

You may think you created that customization exactly the same as in development, but may have missed a few steps in **Test or Production.** You tested your customizations in development and they worked. Now in production they work differently. Trying to diagnose what is different can be pain-staking. If you would have used update sets to begin with, this issue would not have occurred.

Developers have additional options with update sets such as:

1. We can create an update set for a specific version of an application.
2. Specify which application tables want to track in update sets.

Update set tables

When we create an update in service now instance each update set is stored in the Update Set **[sys_update_set]** table, and the customizations that are associated with the update set, which are entries in the Customer Update **[sys_update_xml]** table, appear as a related list on the update set record.

When we want to tracked object is customized, a corresponding record is added or updated in the Customer Update **[sys_update_xml]** table and is associated with the user current update set. The associated application file properties are tracked and transferred along with the customized object in a single update record. A corresponding record is also added to the Versions **[sys_update_version]** table.

Do not directly modify Customer Changes in [**sys_update_xml]** records. It is not best practice

Customizations tracked by update sets

Update sets can track customizations to application tables, fields, and records.

Update sets track customizations under these conditions:

1. Where the table has an **[update_synch]** dictionary attribute.
2. Where there is a special handler to track changes to multiple tables.
3. Where the administrator has not excluded a field from updates.

- ✓ In generaly, update sets capture configuration or customization information but not task or process data.
- ✓ For example, update sets track service catalog item definitions and related configuration data like variables and variable choices.
- ✓ However, if you test the service catalog by placing orders, update sets do not track order requests, items, and catalog tasks.
- ✓ Update sets have a limited capacity to transfer data as application files.
- ✓ For larger data transfers, export data and import it with an import set or web service.

Special handlers

Some changes require special handlers because they represent information on multiple tables. These changes are packaged into one update set entry so that all records are properly updated when the customization is committed. The following changes are tracked with special handlers

- Workflows
- Form sections
- Lists
- Related lists
- Choice lists
- System dictionary entries
- Field labels

What update set Captured

- Choice lists
- Reports
- Workflows
- Form sections
- Lists
- Related lists
- Choice lists
- System dictionary entries
- Field labels
- Tables
- Fields
- Form layout
- List layout
- Business rules, Client scripts, Script includes, UI Scripts, and other scripts
- System properties

What it does NOT capture

- Incidents
- New Data Record
- Modified CI
- Users
- Groups
- Schedules
- Schedule Jobs

Default update set

The new default update set has to do with your application scope. If your new custom application is a privately scope application (it probably is) it will automatically create a new default update set specific to that application. The reason for this is Service Now is always tracking changes to an update set, but you can't have global updates and privately scoped updates in the same update set. You should treat the new update set exactly like you treat the OOB global default update set and manipulate it at all.

Global default Update Set

Use the global default update set to make changes to an instance without adding the changes to any user-created update sets. The global default update set is the set where **Default set = true** and application scope is global. The global default set (regardless of the Name of the set) provides system functionality and should not be **changed, deleted, or moved** between systems. Use this update set to make changes to an instance without adding the changes to any user-created update sets.

Auto-generated default Update Set

At all times, to ensure that no updates to an instance are lost, the system ensures that there is a default set for the user's current scope. If the system finds that a default update set does not exist **(or is marked Ignored or Completed)** for the current scope, then the system auto-generates an update set and sets **Default set = true.**

These are some common cases where the system auto-generates a default update set.

1. The first time that an admin logs into service now instance, the system sets the system's global default update set as the administrator's update set. In addition, the application picker sets the administrator's application scope to global.

2. If a global default update set does not exist (or is marked **Ignored** or **Completed**), the system creates a new update set for the global application scope and performs the following actions:

3. The system sets **Default set = true** for the new update set.

4. When a user marks the default set for a scope as **Ignored** or **Completed** (not a recommended practice), the system immediately auto-generates a new default set for the scope.

The system auto-generates a new default update set for a scope when all the following conditions meet:

1. When we change application scope.
2. When we preferred update set is **Complete or Ignored.**
3. There is no **In-Progress** default update set for the new scope

Update set administration

Service now administrators can configure options for update sets, such as excluding certain fields from updates and controlling access to the update set picker.

Administrators have the following options with update sets:

- Create an update set to store local changes.
- Select the current update set to store local changes.
- Commit an update set to prepare it for distribution.
- Report on the contents of update sets.
- Compare update sets to determine what differences they contains.
- Merge separate update sets into a single update set.
- Create an external file from an update set.
- Retrieve update sets from remote instances.
- Apply retrieved update sets.
- Back out changes applied from an update set.
- Set system properties related to update sets

Track an application table

Application developers can track application changes in an update set to save or distribute a particular version of an application. During table creation, set Extends Table to Application File **[sys_metadata].**

Grant access to the update set picker

The update set picker allows users to choose an update set for making and tracking customizations. By default, only administrators can use the update set picker. You can grant access to additional users.

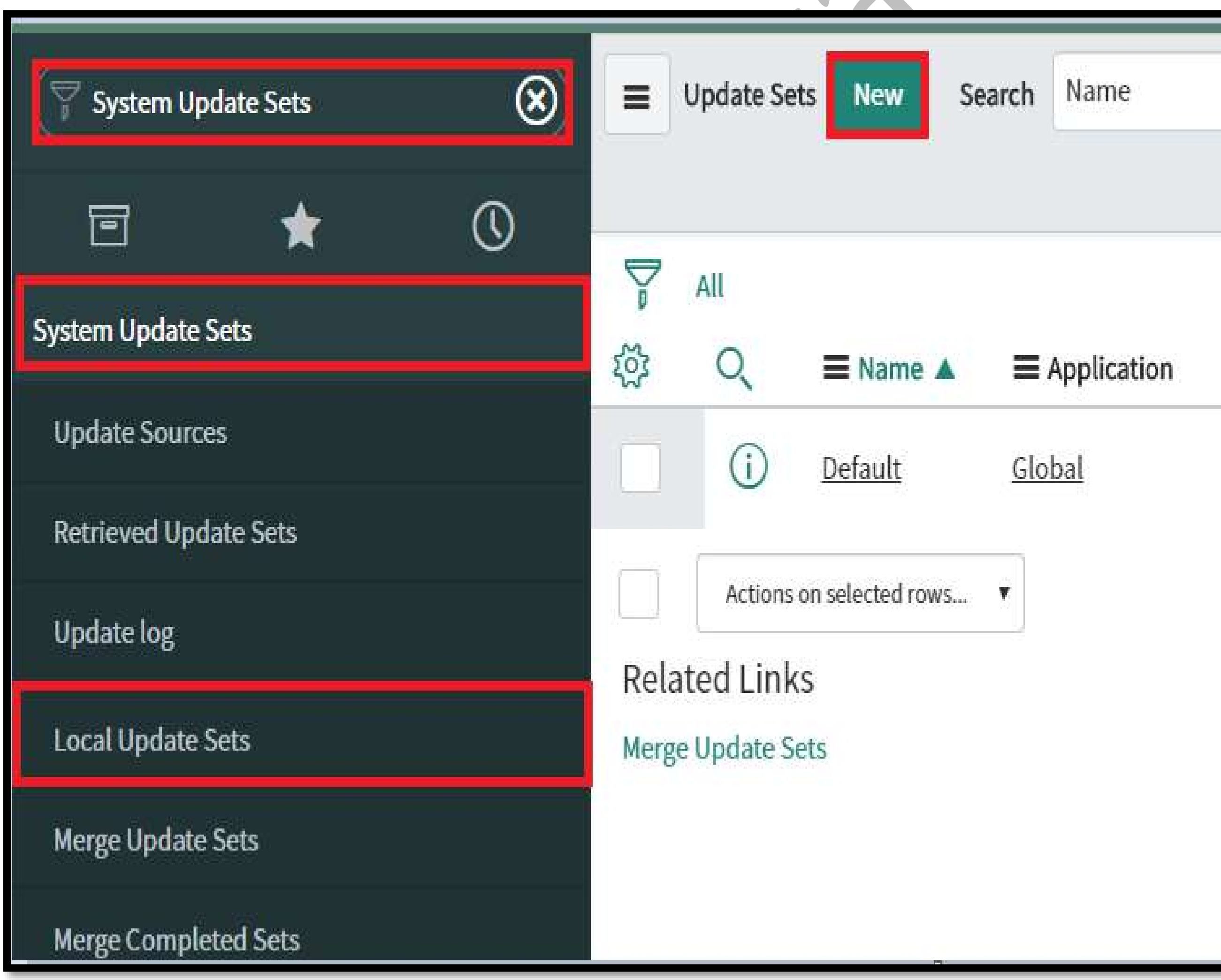
Exercise 1: Create an Update Set

An update set is a group of customizations that can be used to move changes from one instance to another instance.

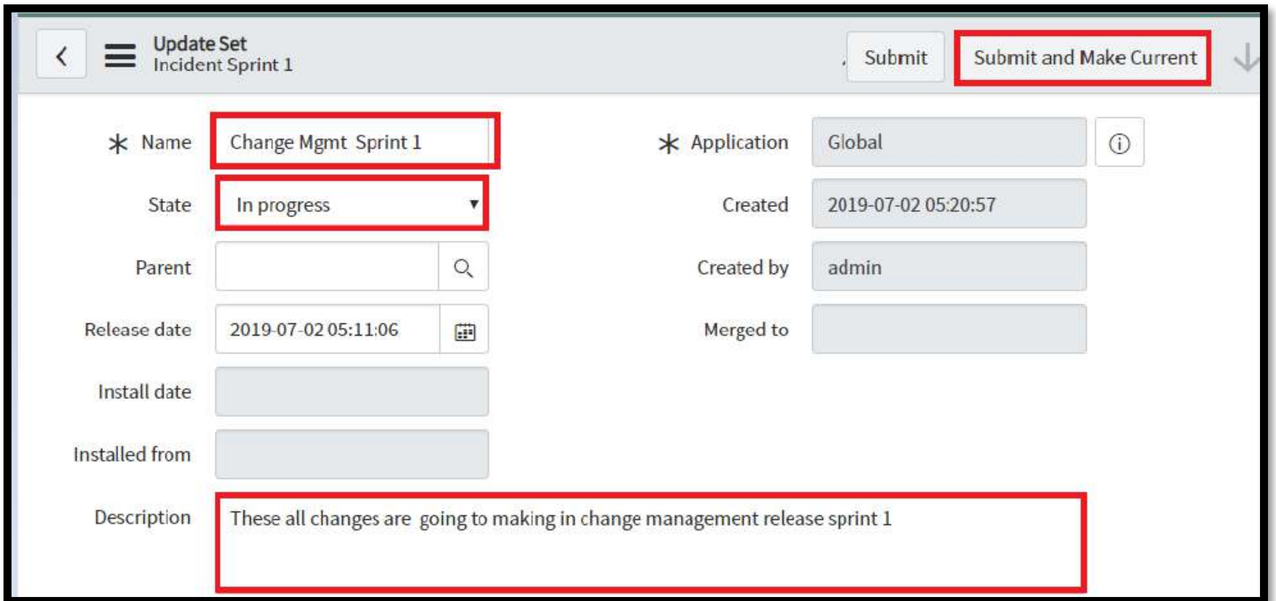
When we created an update set then it can be marked as the current update set to all changes are tracked in it.

Procedure

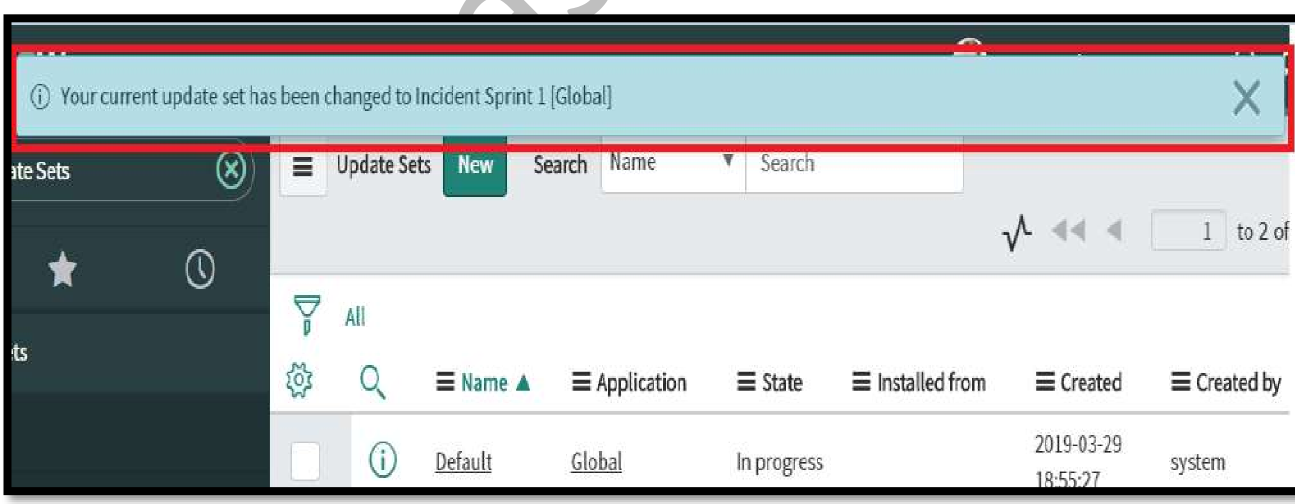
1. Navigate to **System Update Sets > Local Update Sets**
2. Click on **New.**



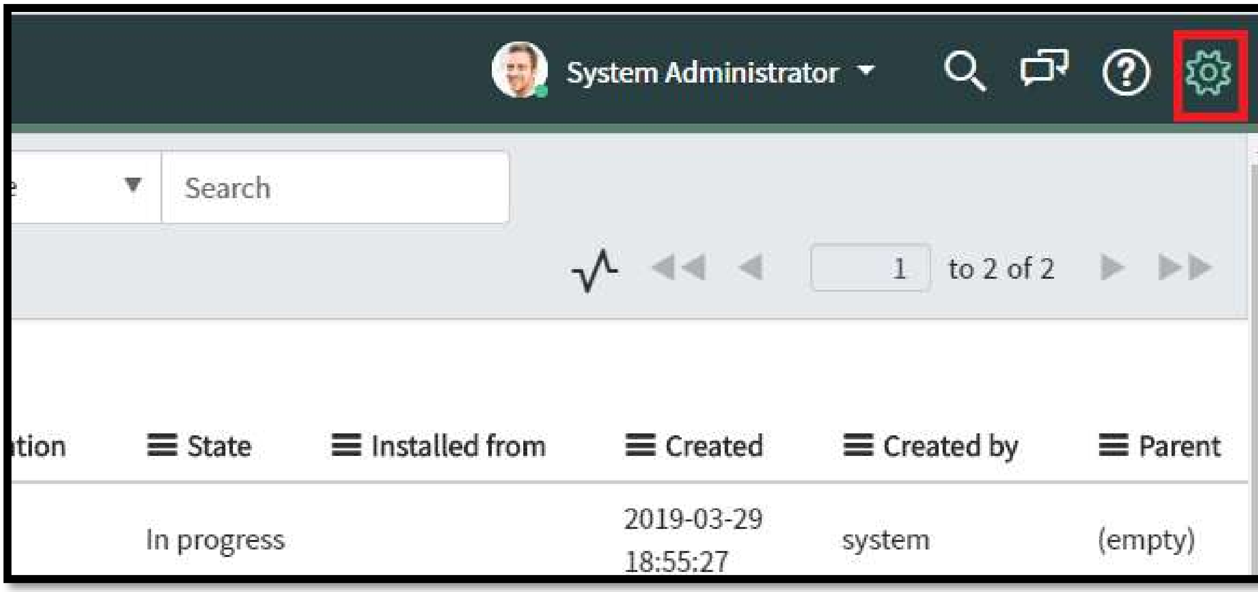
3. Enter **Update Set** details as described in the table.



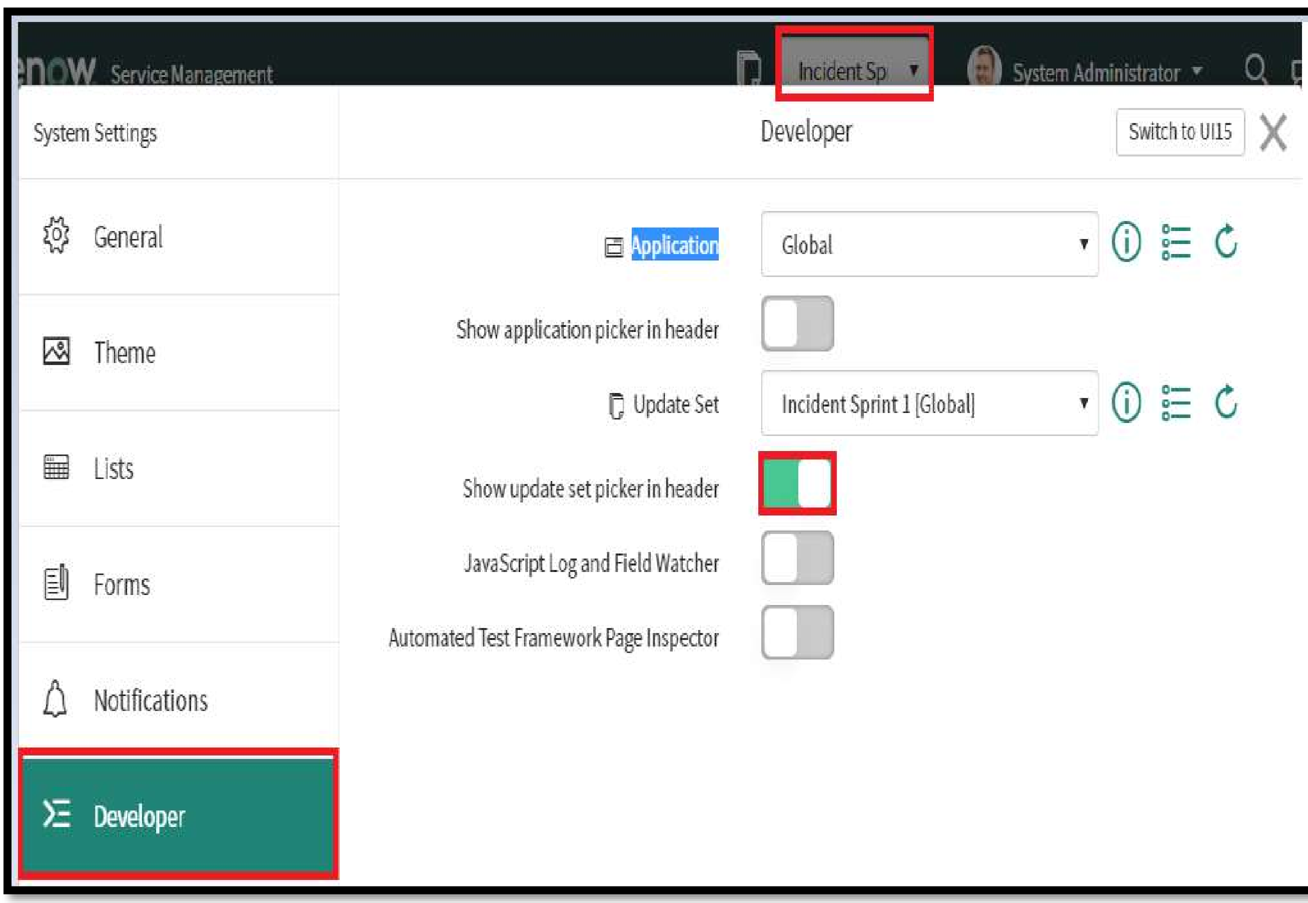
4. **Name:** Change Mgmt Sprint 1
5. **State:** In progress **(Default)**
6. **Release date:** Optional
7. **Description:** These all changes are going to making in **Change Mgmt** release sprint 1
8. Click on **Submit and Make Current**



9. Find above screen shot for reference
10. Select the new Update Set in the Update Set picker in your instance.



11. Go to **Banner Frame** click on **Settings Icon**



12. Select **Developer** tab
13. Show update set picker in header: Checked true

Done Customizations in Change Management Application

We are going to making some changes in **Change Management Application** based on client requirement

Expecting Changes

Story 1: Form validation and Restrictions

Mandatory: Short description, requested by, Requested by date, Change Plan

Read Only: Number Field

Visibility: In state **"Closed"** the fields Assigned to and Assignment group should not be seen

Story 2: Add Related List Affected Locations to

As a user, I want to see – as a list – which locations are affected by a change looking at its current form.

Story 3: Create Data Policy to restrict user from server side

When state is **Closed,** then **Assigned to** and **assignments group** fields should be hide

Story 4: Create New Section add some fields into that

Create **More Information section** and added some fields such as, **closed, closed by, created by, created, updated, updated by**

Story 5: Add one more choice to Category field

Add a choice to **Category** field that choice is **HR**

Exercise :2 Working with Story 1

Story 1: Form validation and Restrictions

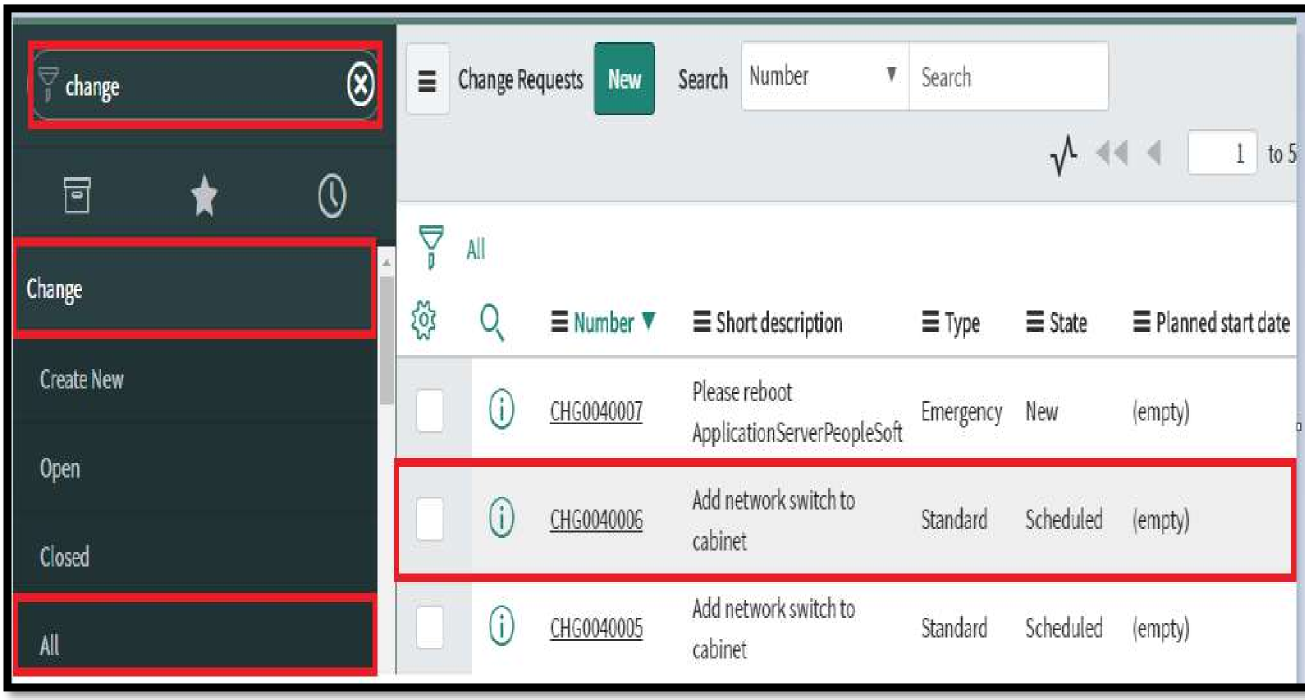
Mandatory: Short description, requested by, Requested by date, Change Plan

Read Only: Number Field

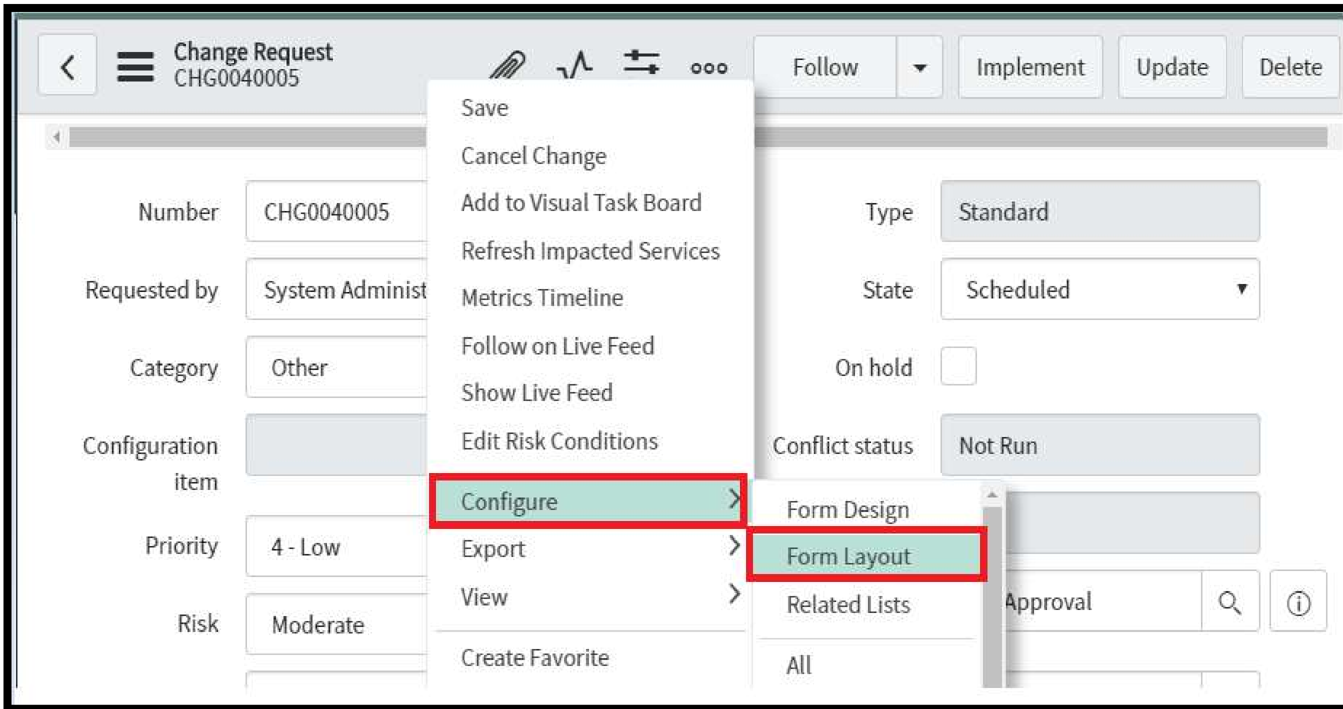
Visibility: In state **"Closed"** the fields Assigned to and Assignment group should not be seen

Procedure

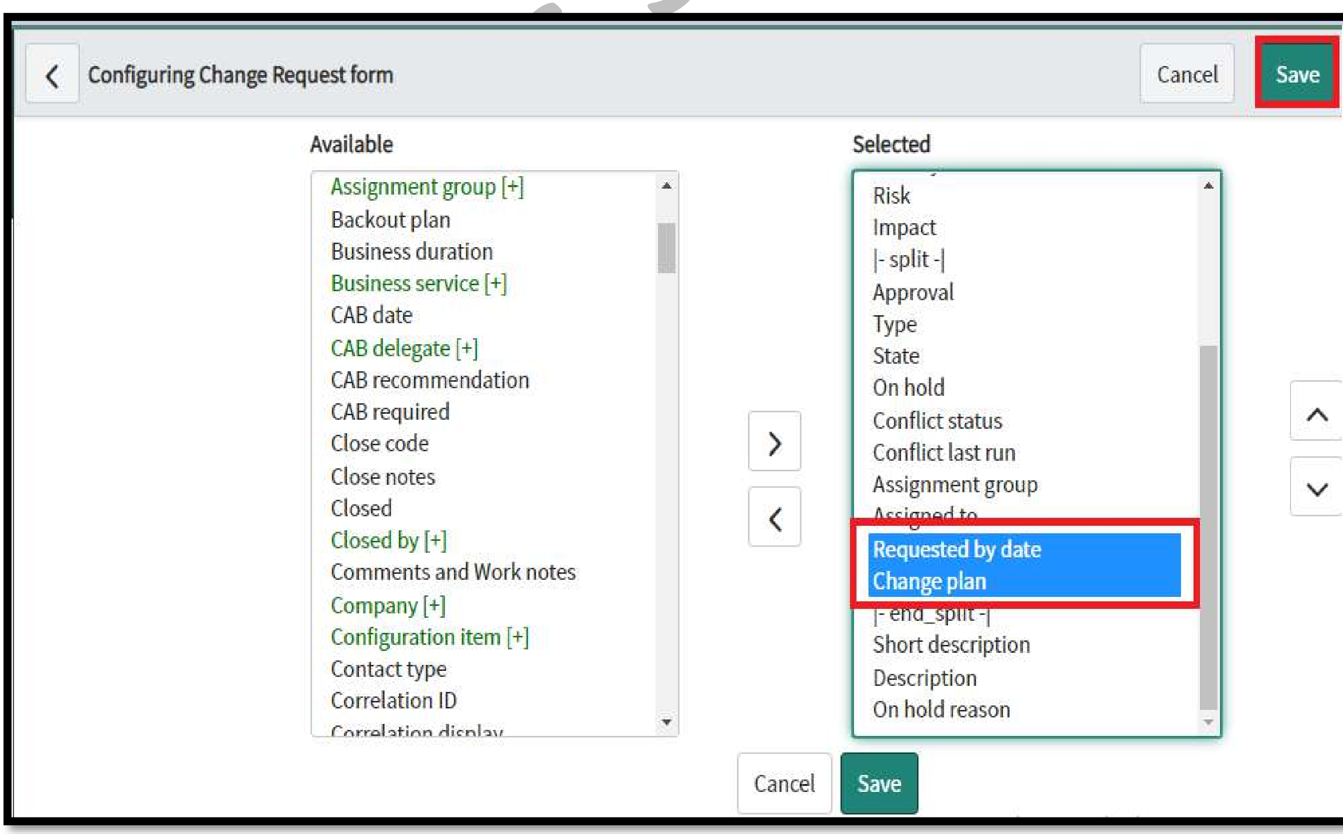
1. Navigate to **Change application > Open existing record**



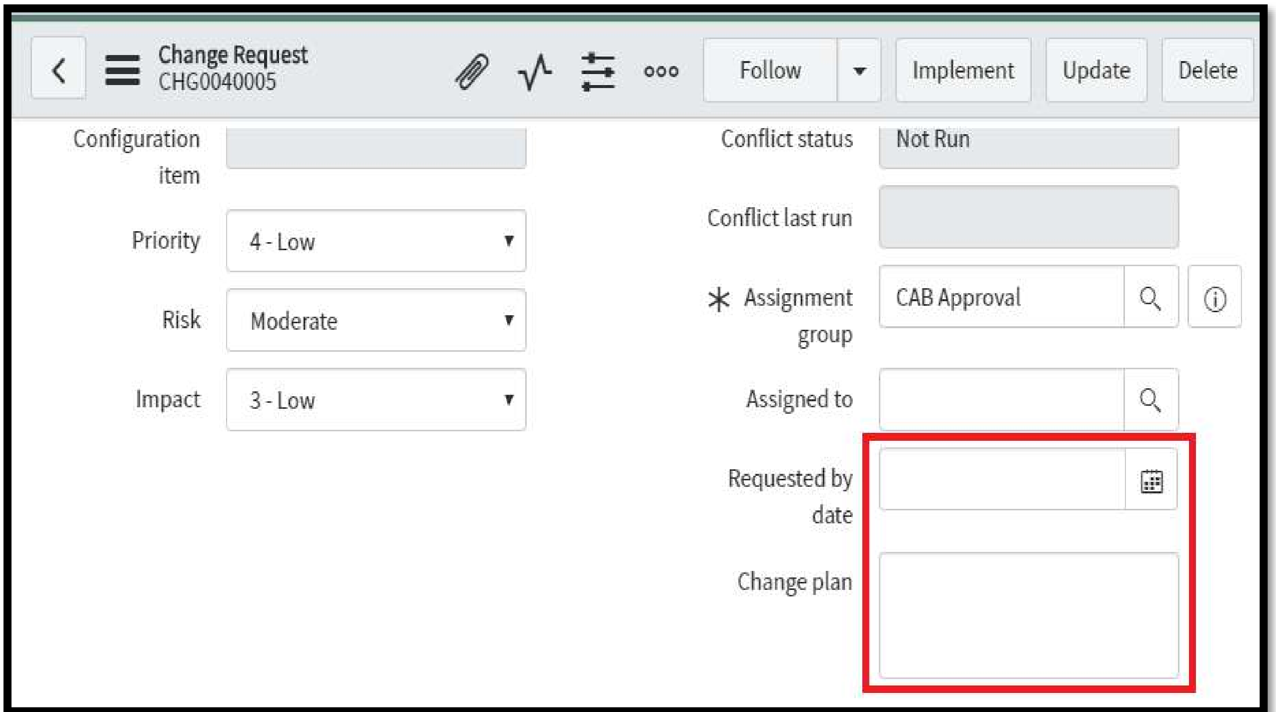
2. First add **Requested by date** and **Change Plan** fields to your form
3. Right click on **Form Header**
4. Navigate to **Configure > Form Layout**



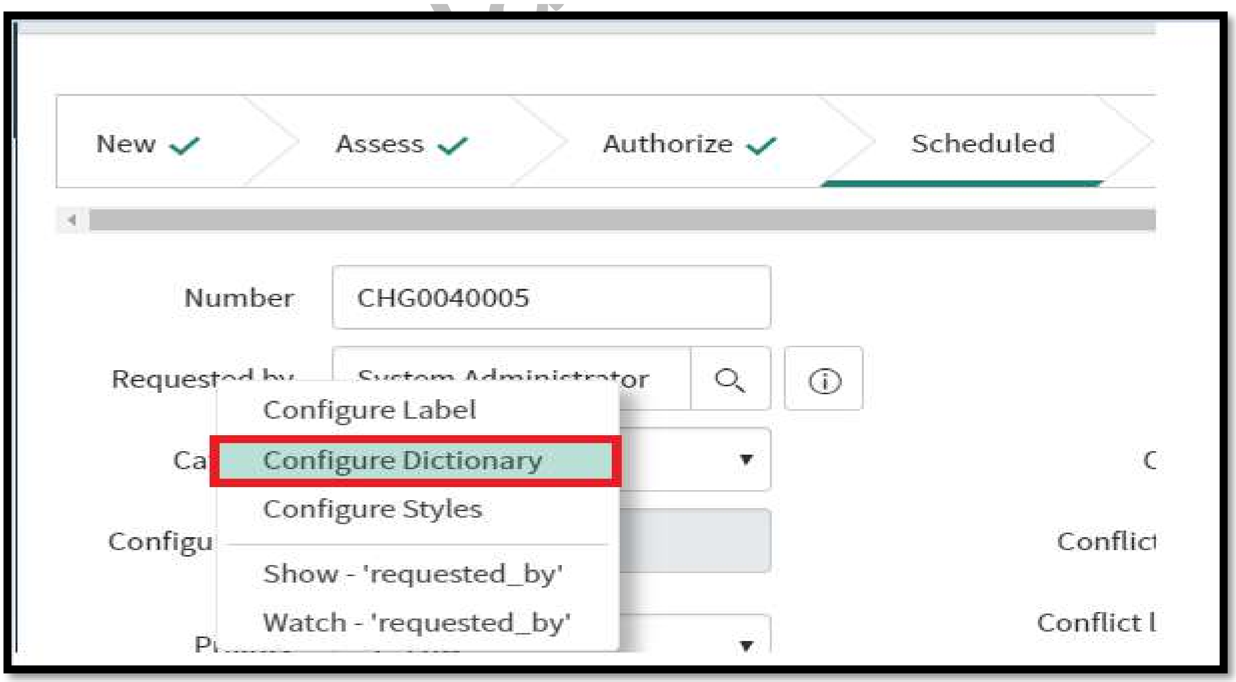
5. Add **Selected fields** to your form
6. Click on **Save**

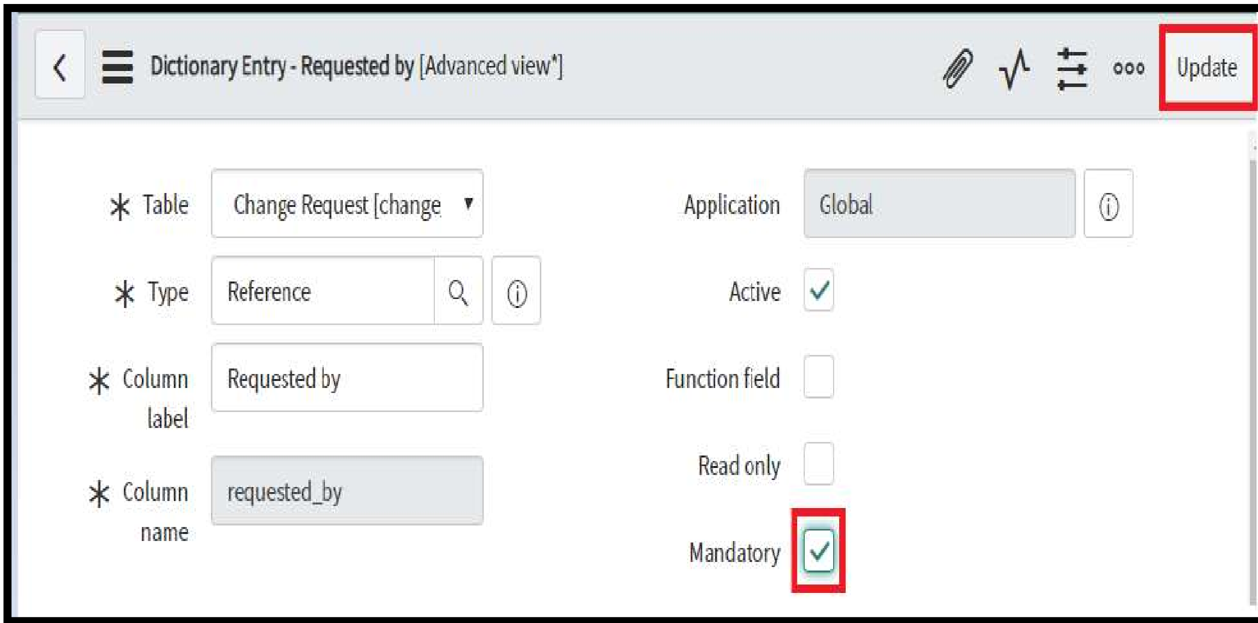


7. Check **your form** after added these fields



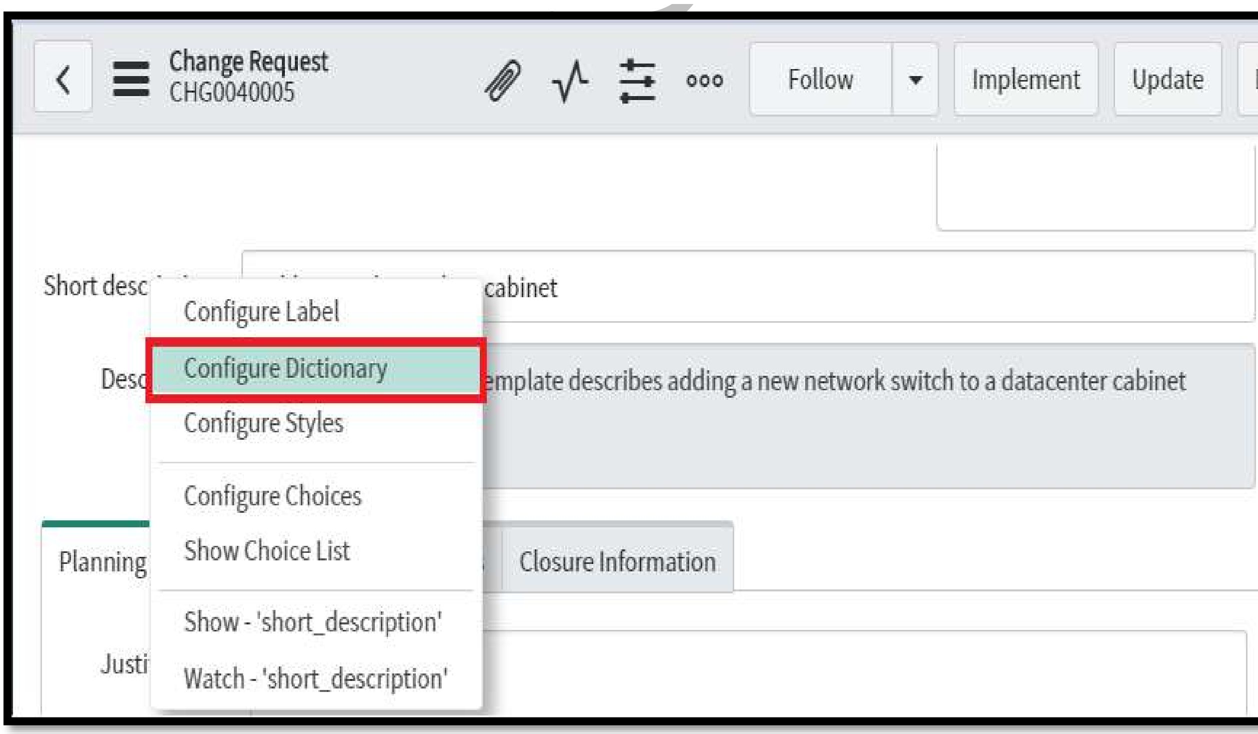
8. **Make Mandatory:** Short description, requested by, Requested by date, Change Plan fields
9. Right Click on **Requested by > Configure Dictionary**



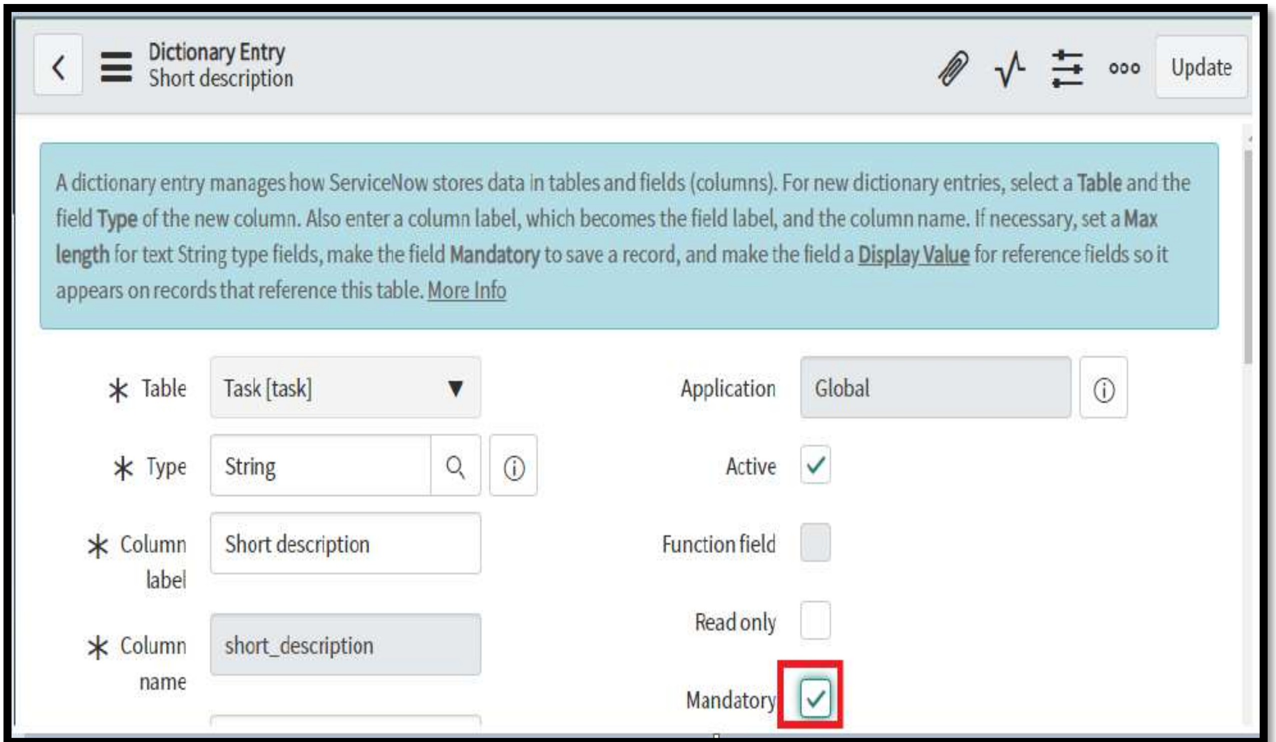


10. Enable **Mandatory** Checkbox
11. Click on **Update**

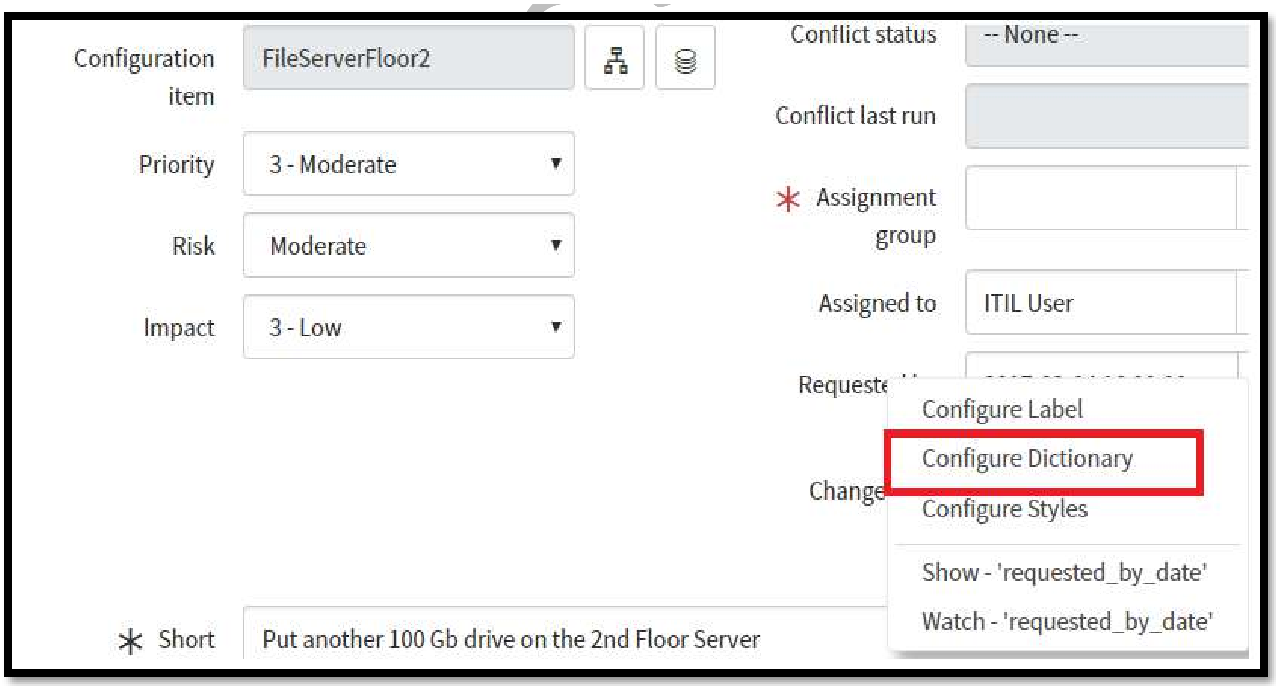
12. Right Click on **Short Description > Configure Dictionary**



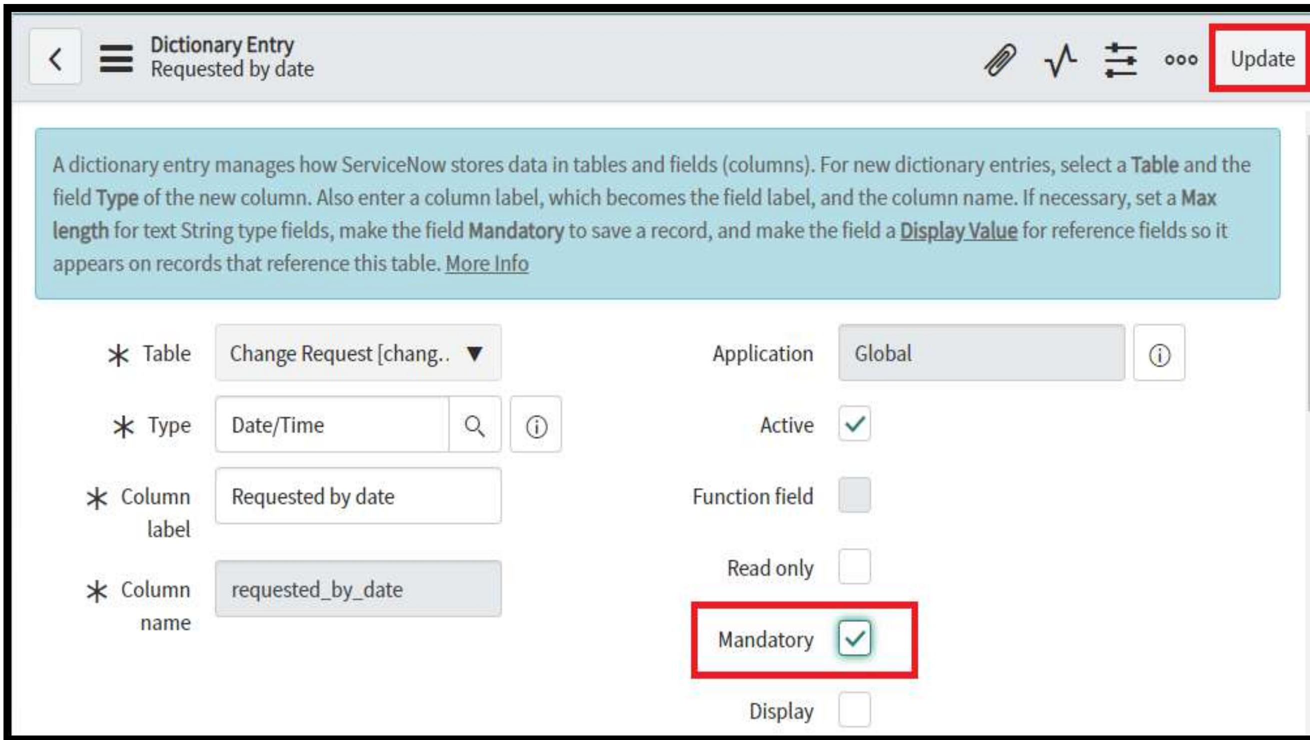
13. Enable **Mandatory** Checkbox
14. Click on **Update**



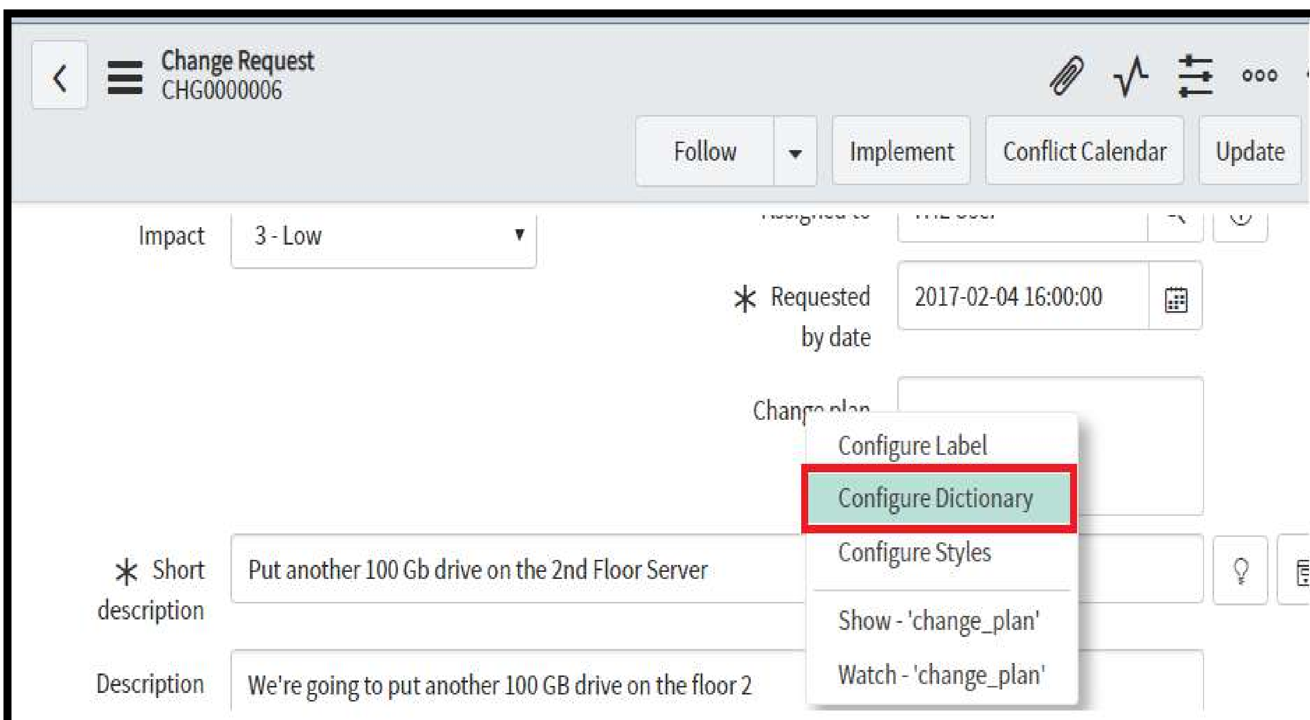
15. Right Click on **Requested by date > Configure Dictionary**



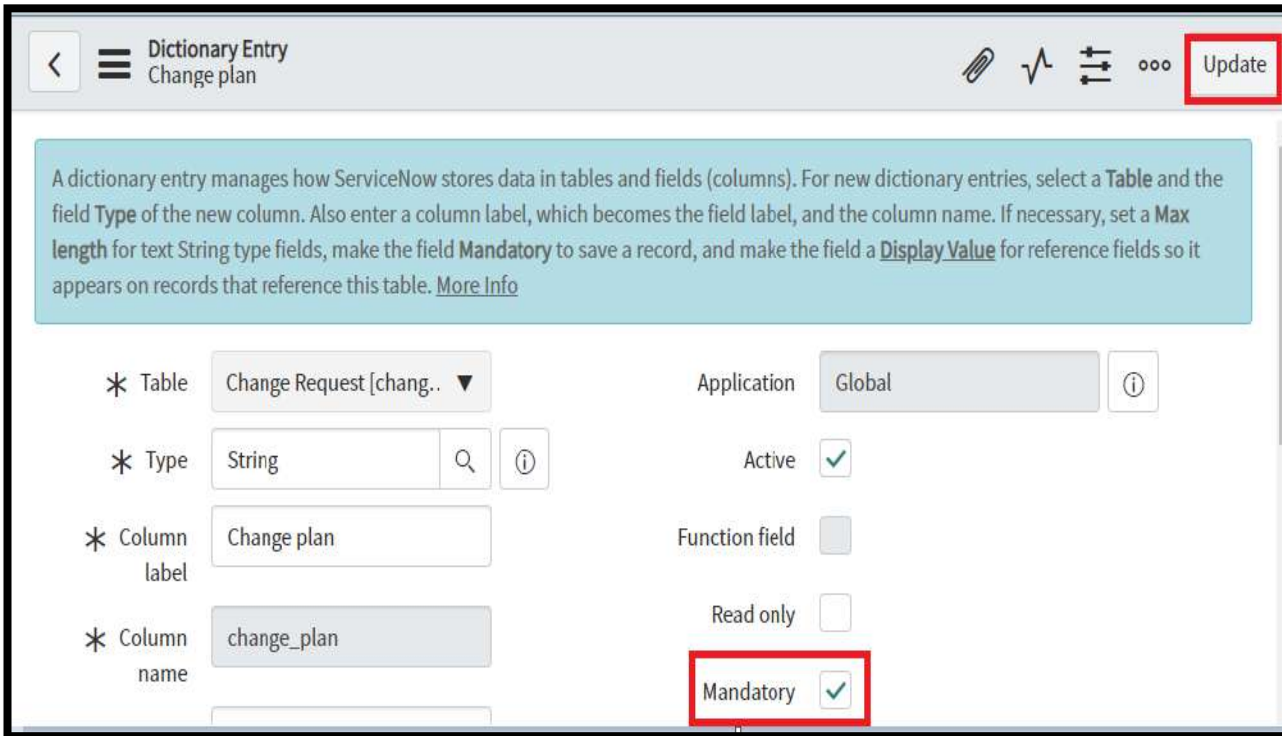
16. Enable **Mandatory** Checkbox
17. Click on **Update**



18. Right Click on **Change Plan > Configure Dictionary**



19. Enable **Mandatory** Checkbox
20. Click on **Update**



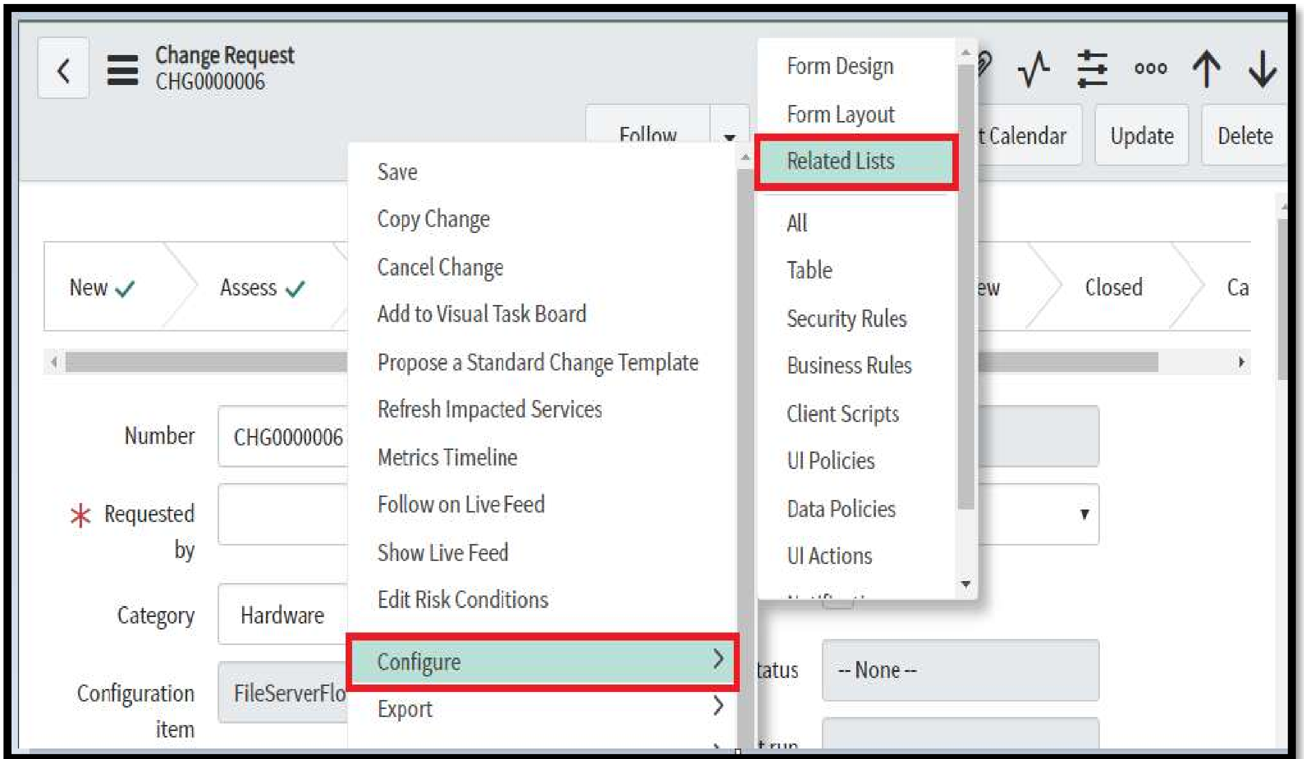
Exercise :3 Working with Story 2

Story 2: Add Affected Locations Related List to Change Application

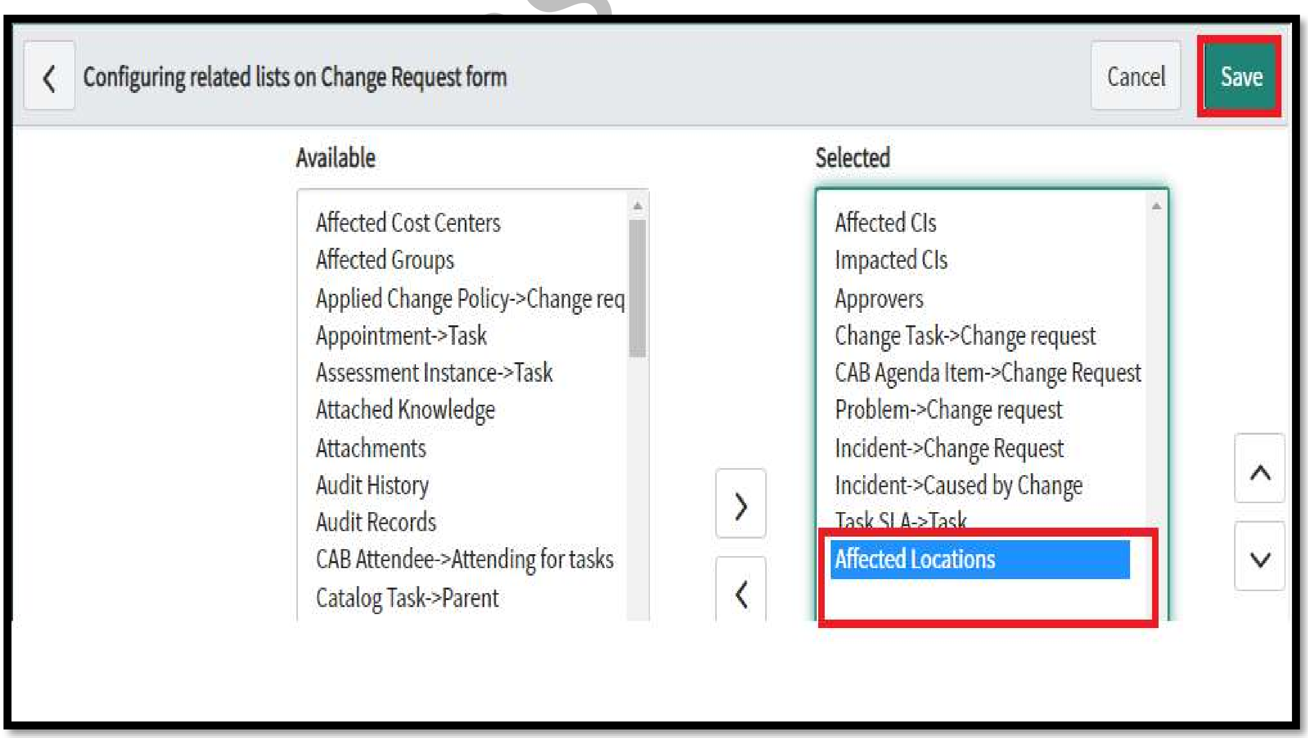
As a user, I want to see – as a list – which locations are affected by a change looking at its current record form.

Procedure

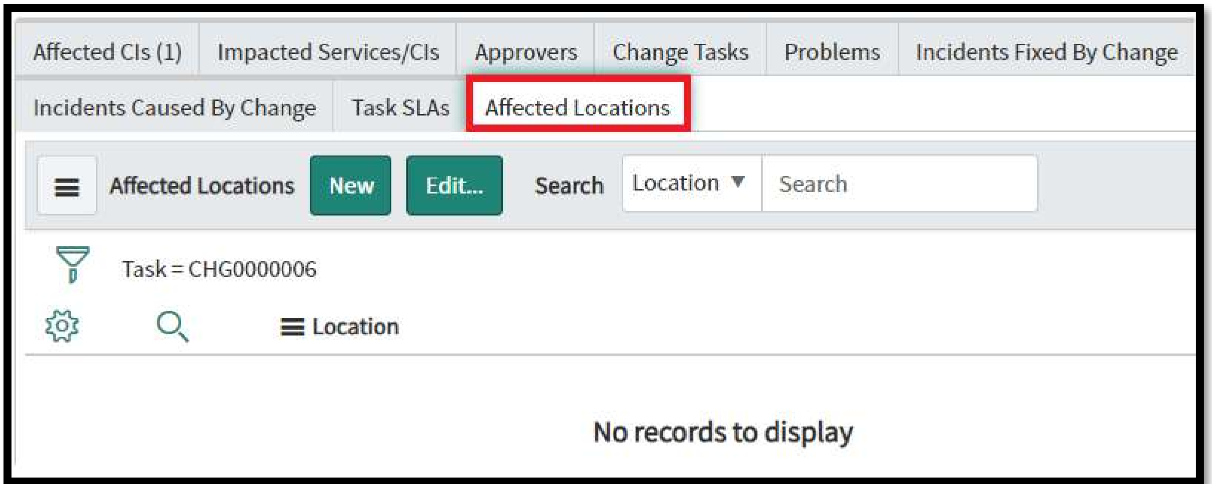
1. Navigate to **Change Application > Open Existing Record**
2. Right click on **Form Header**
3. Configure
4. Related Lists



5. Add **Effected Location** as selected
6. Click on **Save**



7. After Added related list check like below screen shot



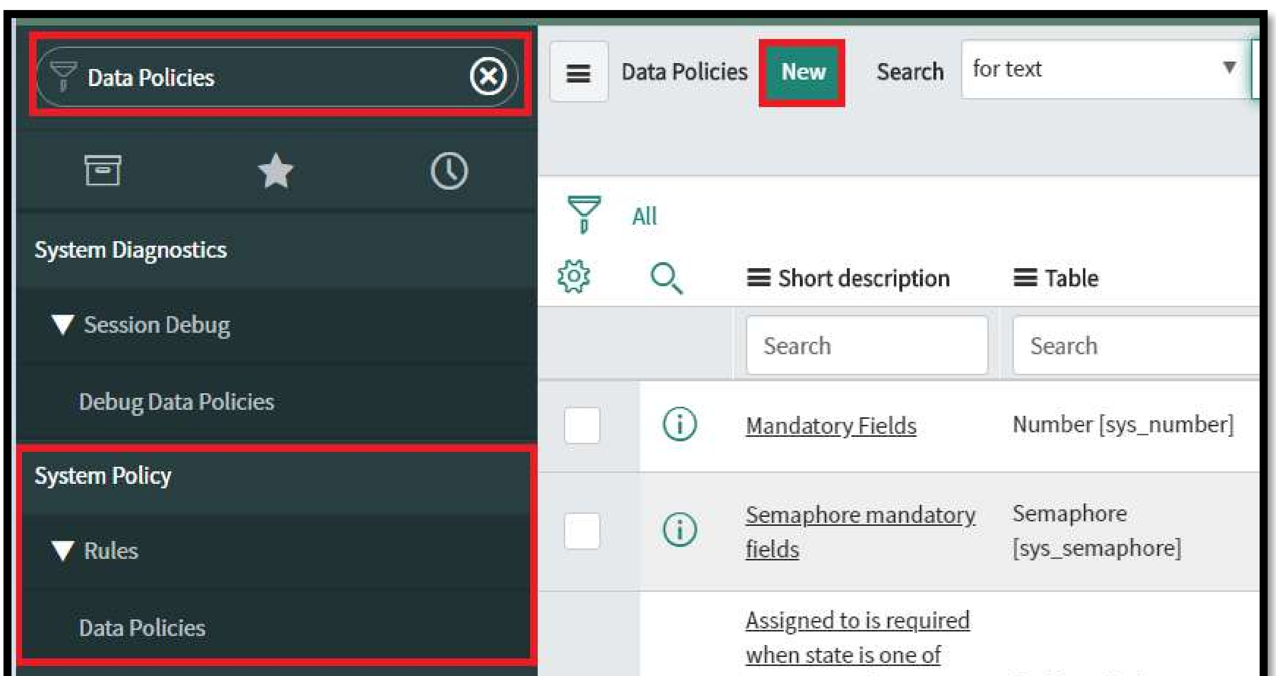
Exercise :4 Working with Story 3

Story 3: Create Data Policy to restrict user from server side

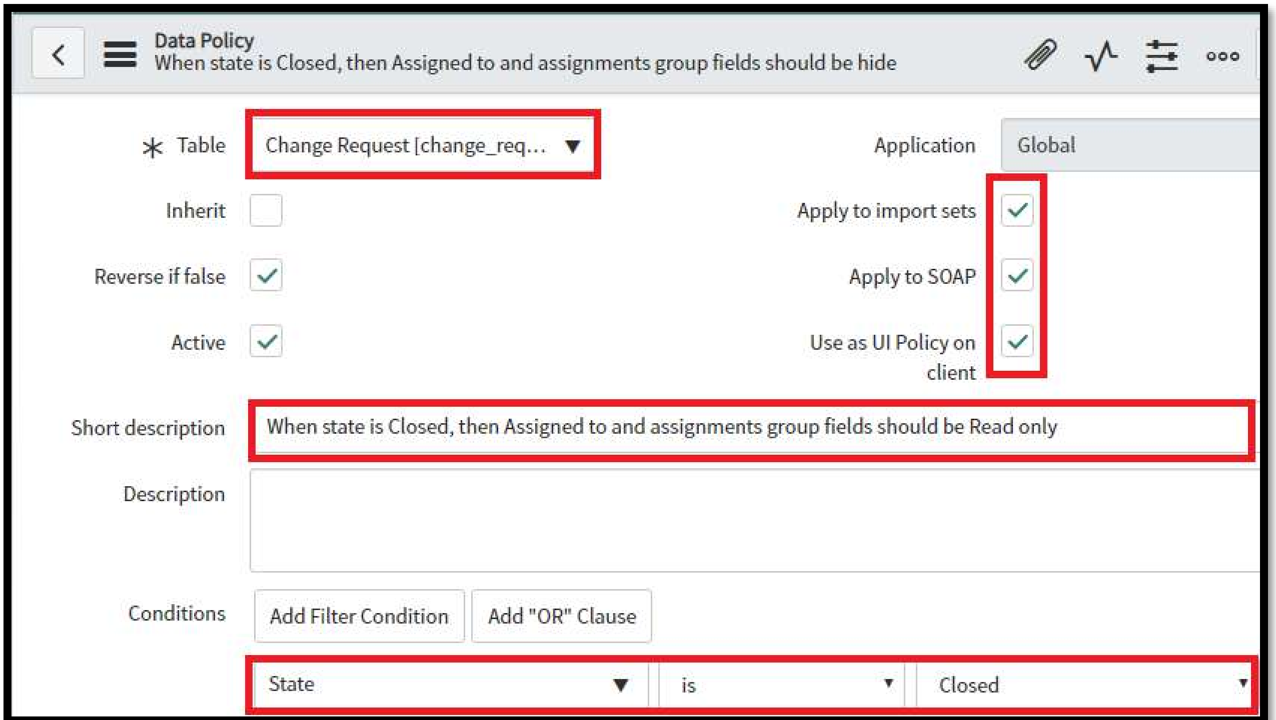
When state is **Closed,** then **Assigned to** and **assignments group** fields should be Read only

Procedure

1. Navigate to **System Policy > Rules > Data Policies**
2. Click on **New**

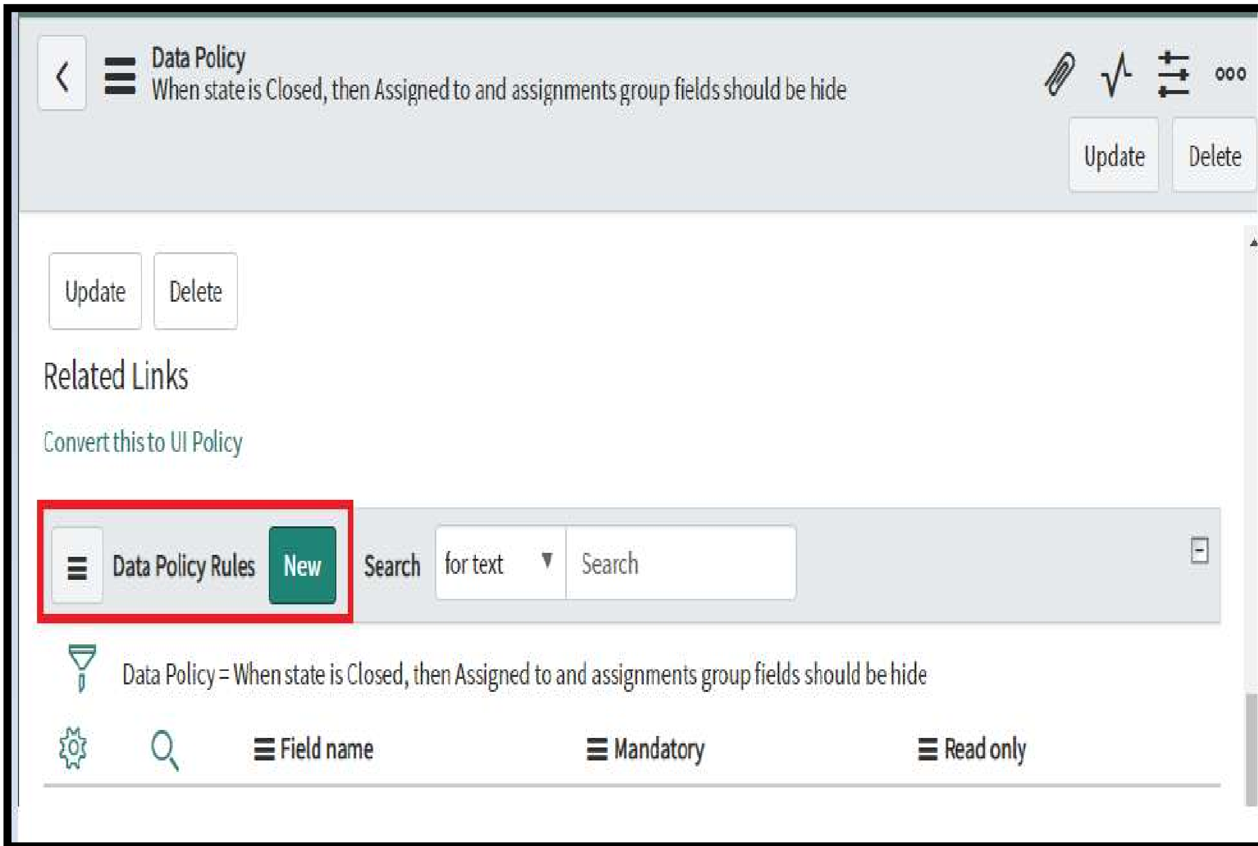


3. Fill **Data Policy** form

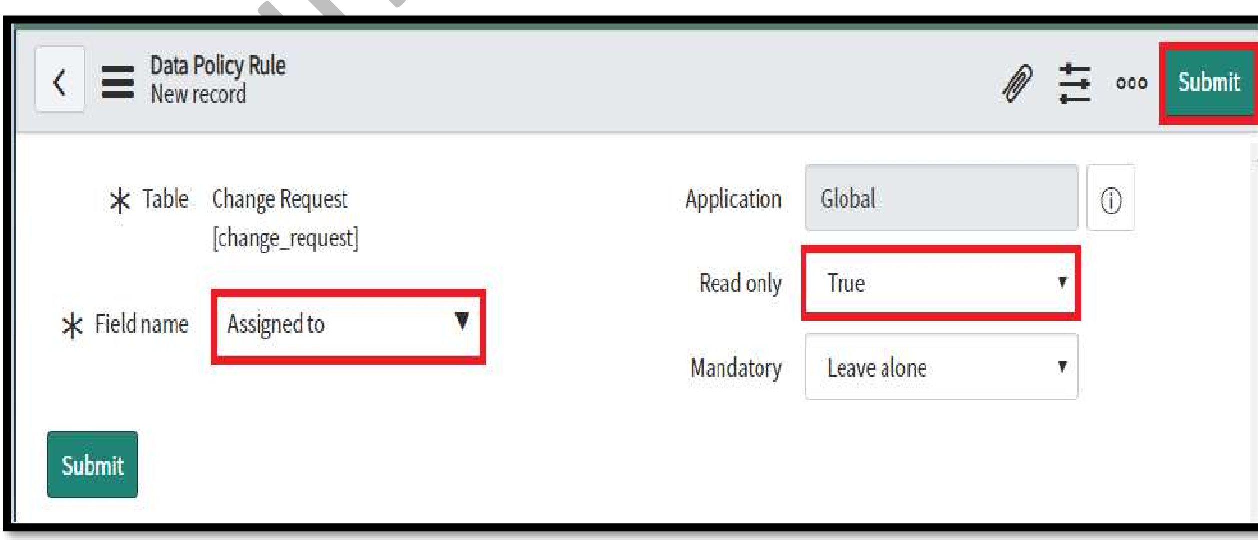


4. **Table:** change_request
5. **Apply to import sets:** Auto Filled
6. **Apply to SOAP:** Auto Filled
7. **Use as UI Policy on client:** Auto Filled
8. **Reverse if false:** Auto Filled
9. **Active:** check box auto filled
10. **Short_description:** When state is **Closed,** then **Assigned to** and **assignments group** fields should be read only
11. **Condition:** [State] [is] [Closed]
12. Right click on **Data Policy form** header and click on **Save**

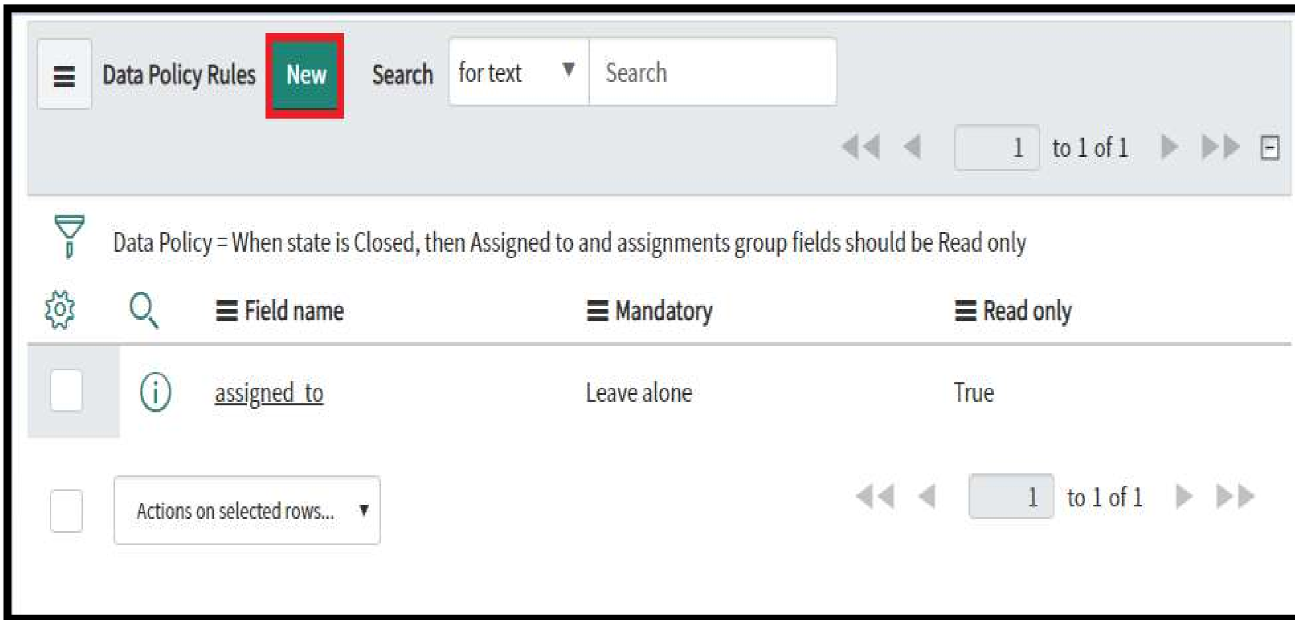
13. Come **down** in the same form
14. Click on Data Policy Rules **New**



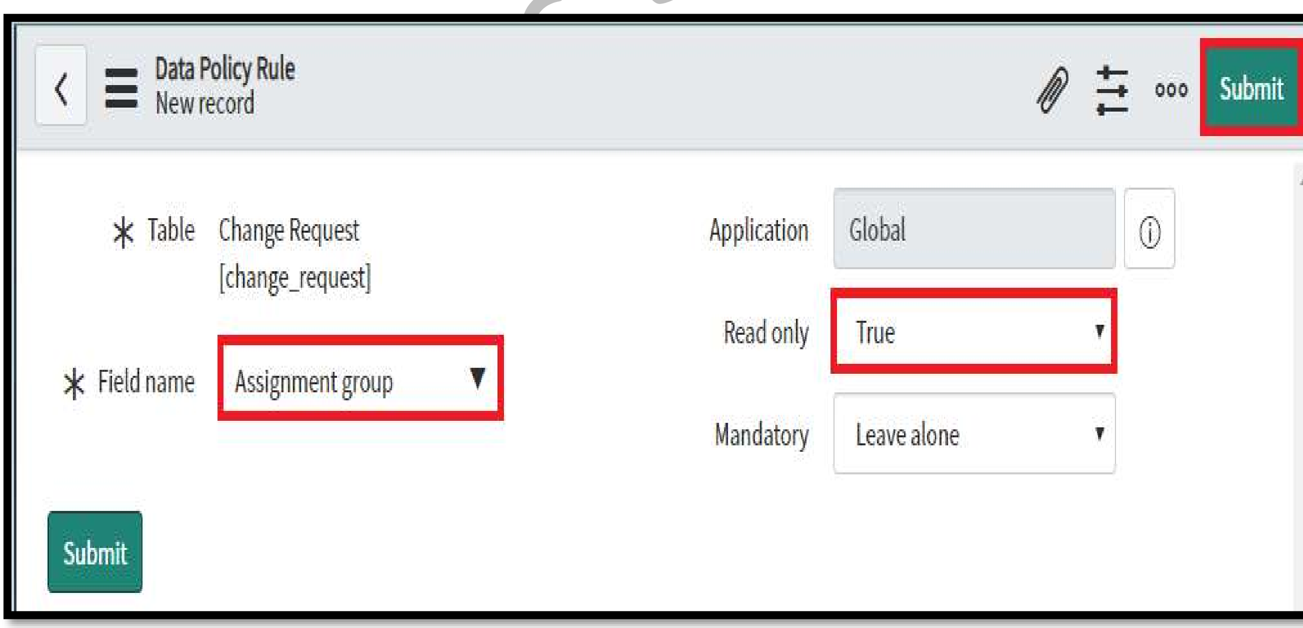
15. Field Name: Assigned to
16. Read Only: True
17. Click on **Submit**



18. Click on Data Policy Rules **New**



1. **Field Name:** Assignment Group
2. **Read Only:** True
3. Click on **Submit**



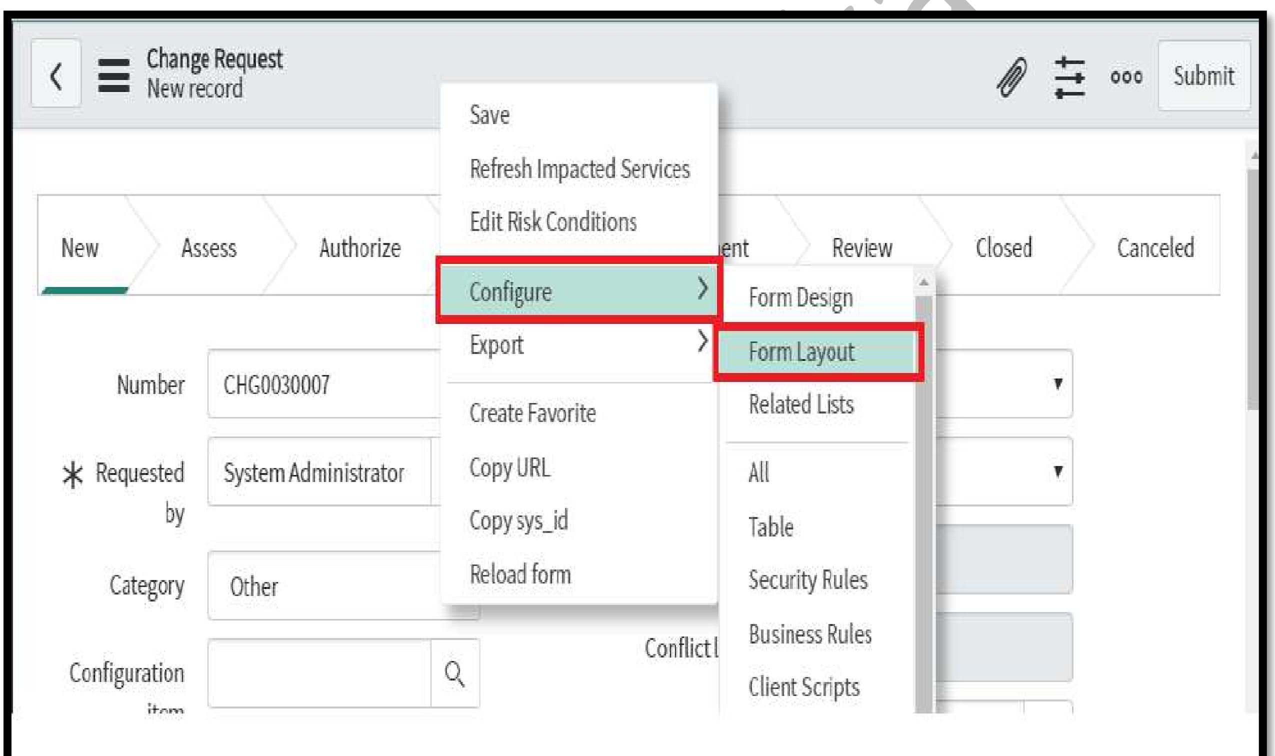
Exercise :5 Working with Story 4

Story 4: Create New Section add some fields into that

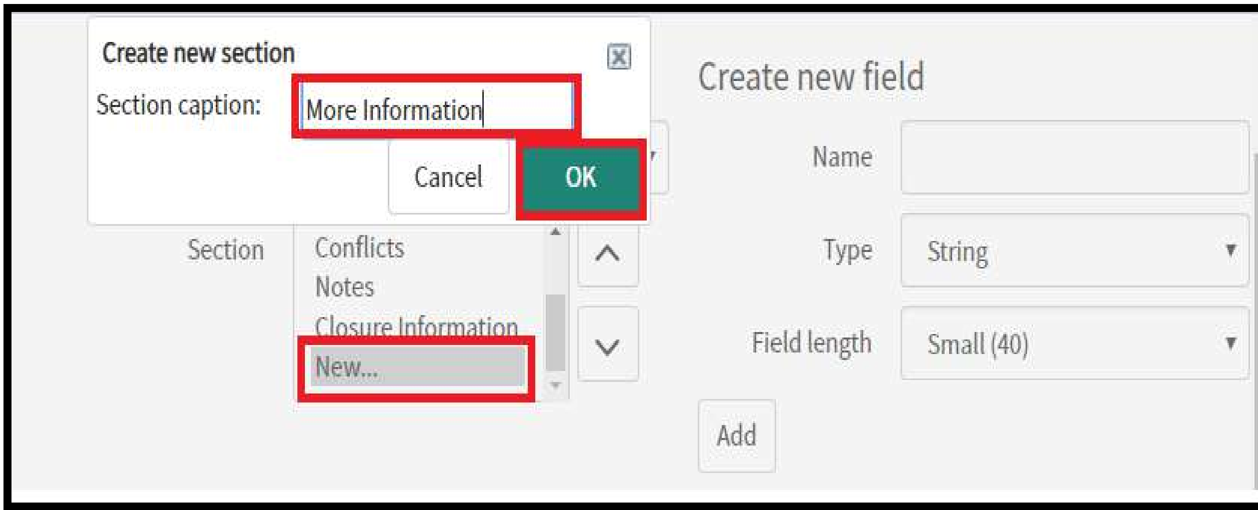
Create **More Information** section and added some fields such as, **closed, closed by, created by, created, updated, updated by**

Procedure

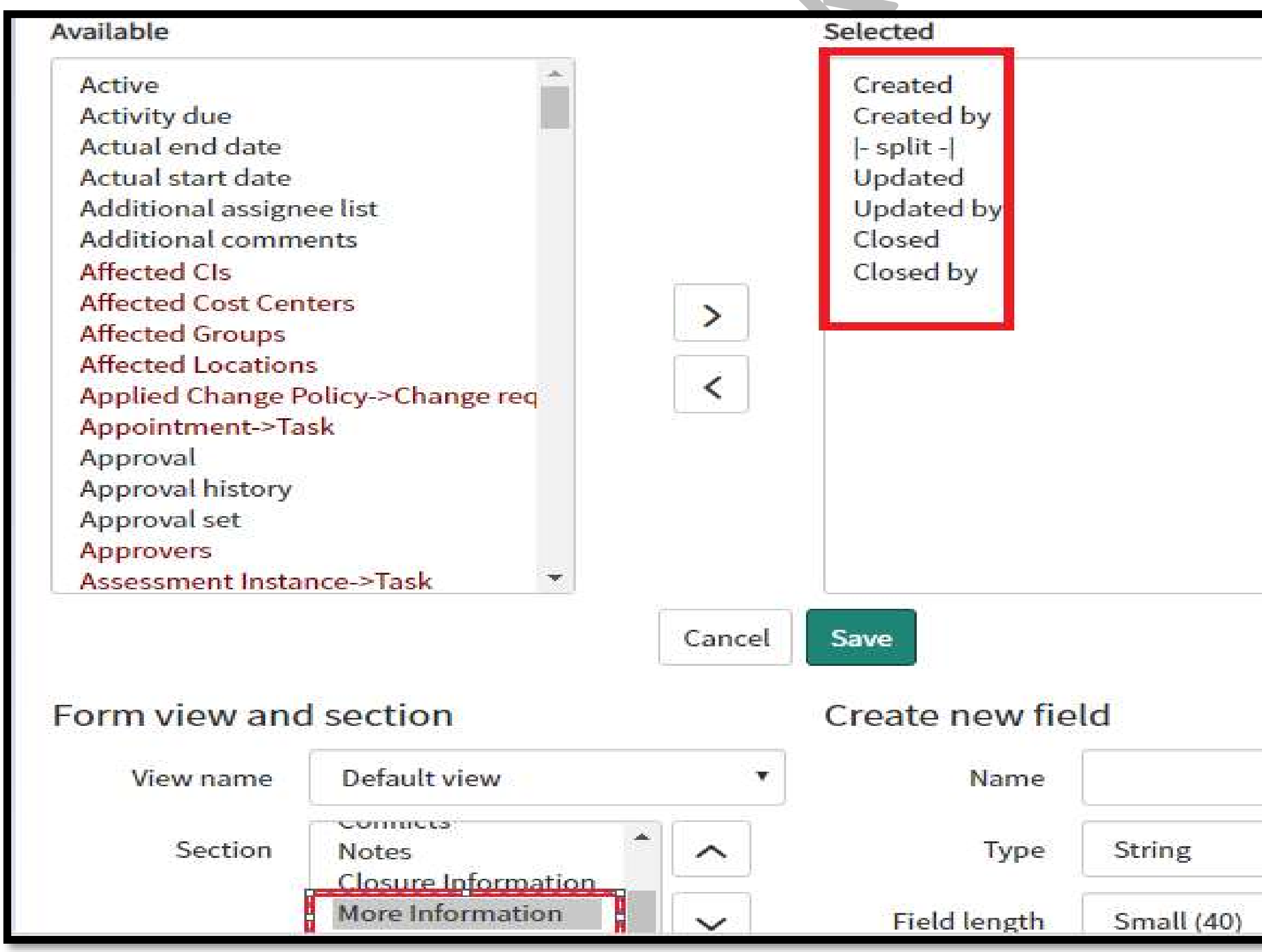
4. Navigate to **Change Application** > **Open existing change record**
5. Right click on **Form Header**
6. **Configure** > **Form Layout**



7. Go to section tab and click on **New**
8. **Section Caption:** More Information
9. Click on **New**
10. Click on **OK**



11. Add these all **Selected** fields



12. Click on **Save**

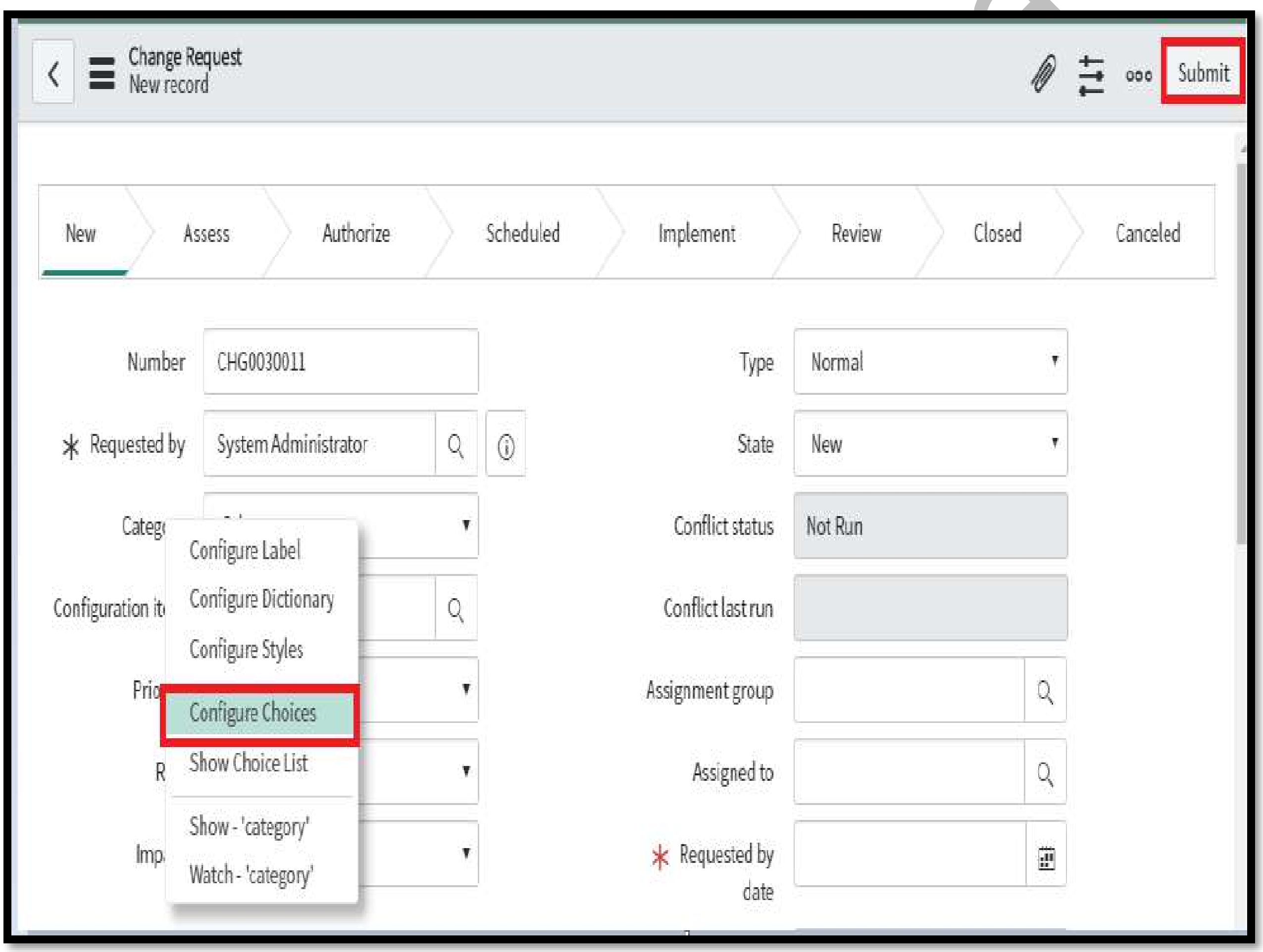
Exercise :6 Working with Story 5

Story 5: Add one more choices to Category field

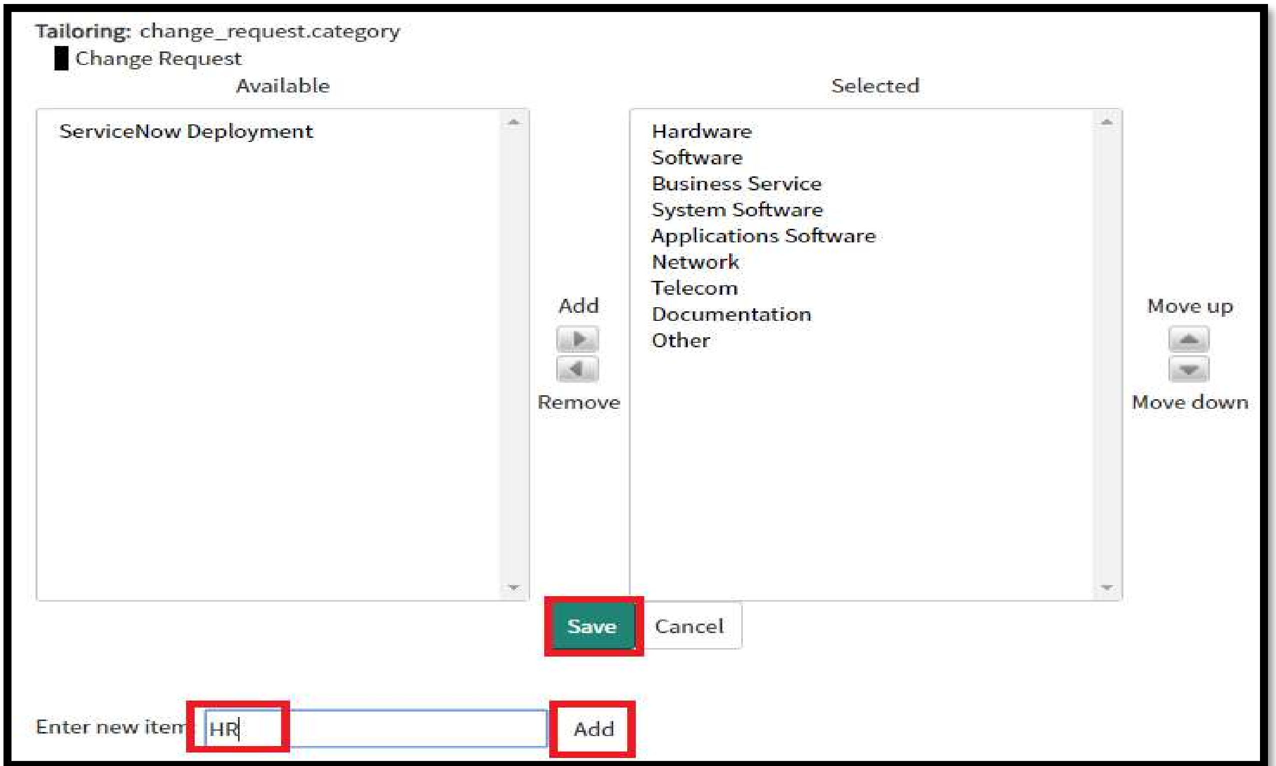
Add a choice to **Category** field that choice is **HR**

Procedure

1. Navigate to **Change Application** > **Open existing change record**
2. Right click on **Category** field
3. Click on **Configure Choices**



4. **Enter New Item:** HR
5. Click on **Add**
6. Again click on **Save**

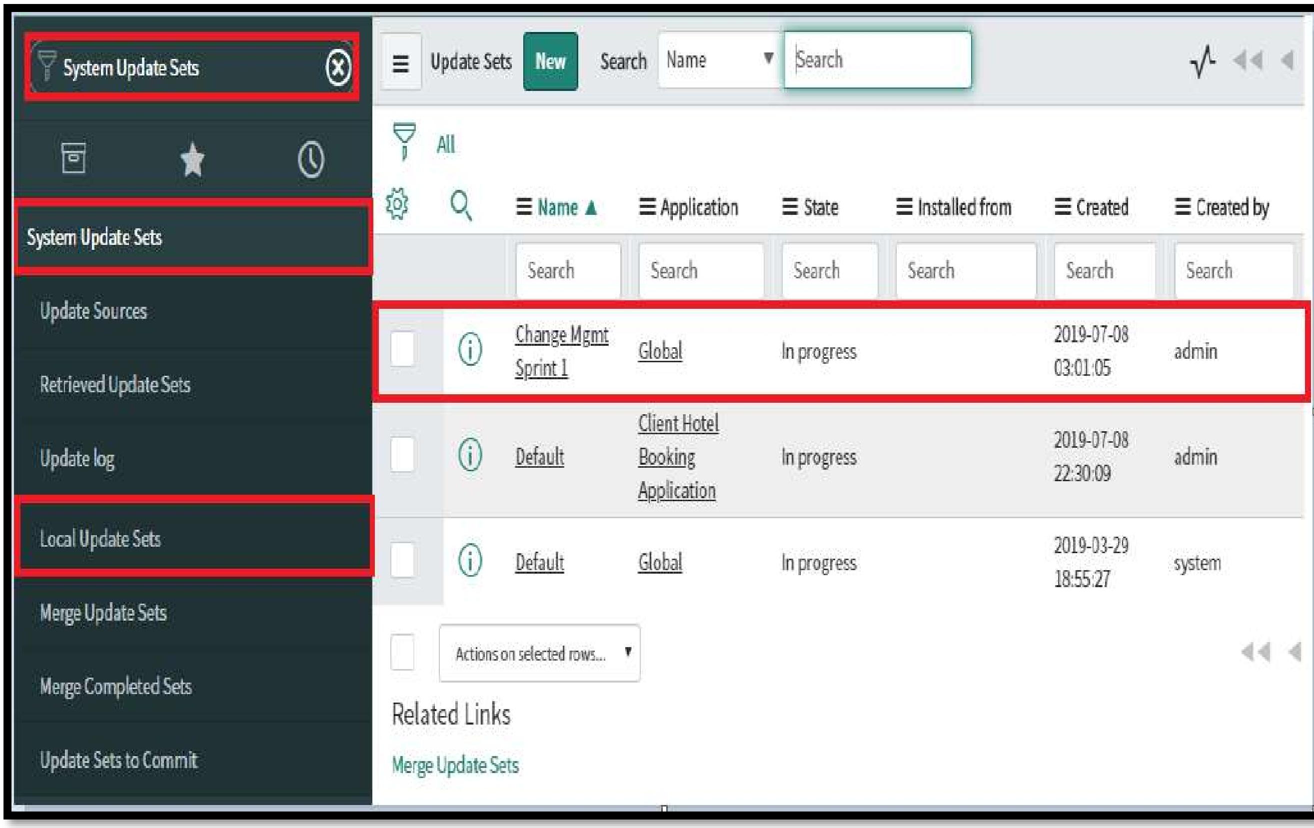


Exercise :7 Export all your changes into XML

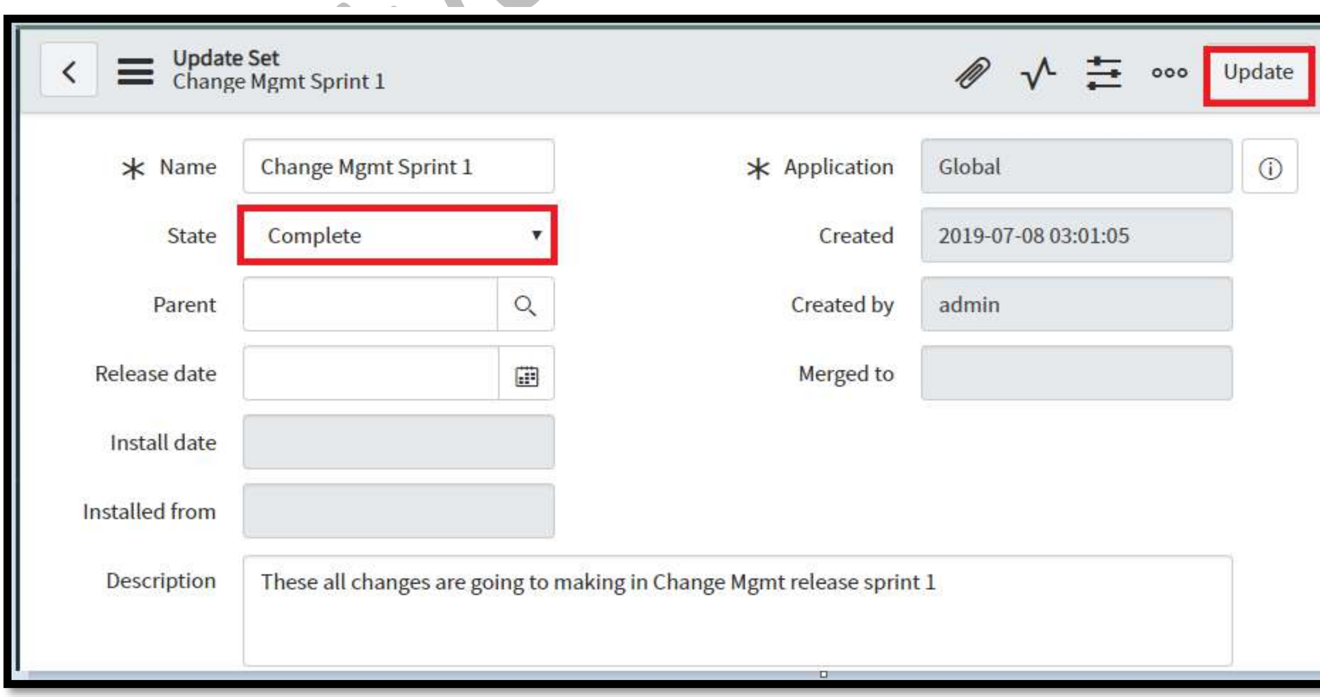
After completed all our changes in **Development Instance** then we can move these changes to **Test Instance**

Procedure

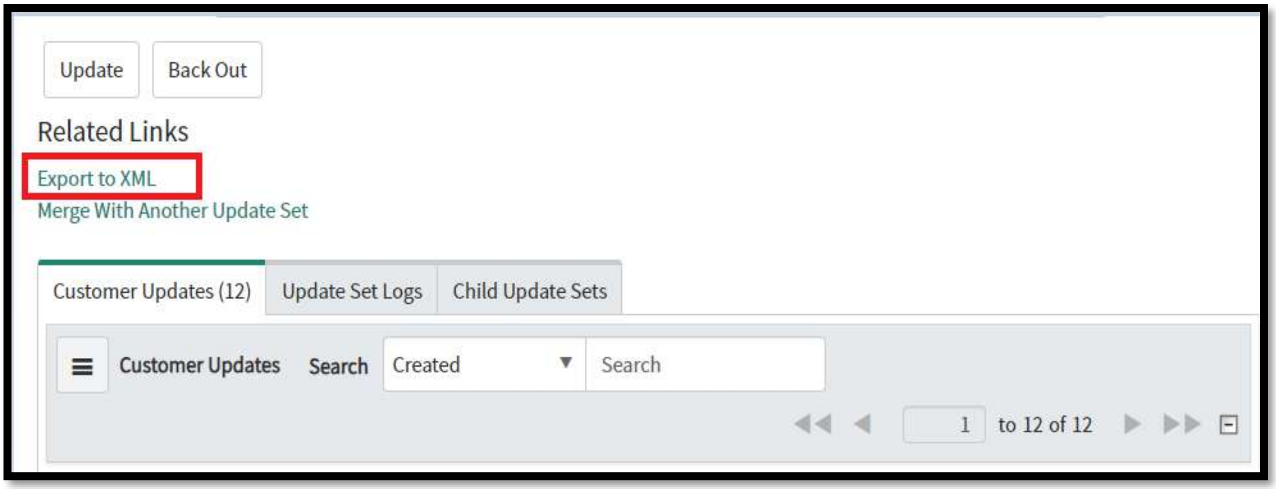
1. Navigate to **System Update Sets > Local Update Sets**
2. Open our **Local Update Sets** record **> Change Mgmt Sprint 1**



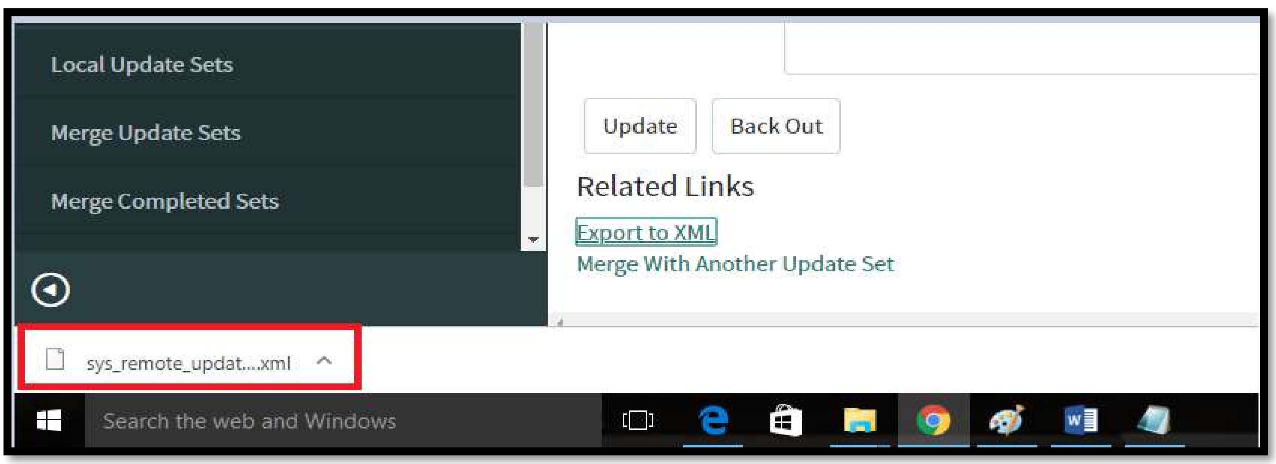
3. Change State value from **In progress** to **Complete**
4. Click on **Update**



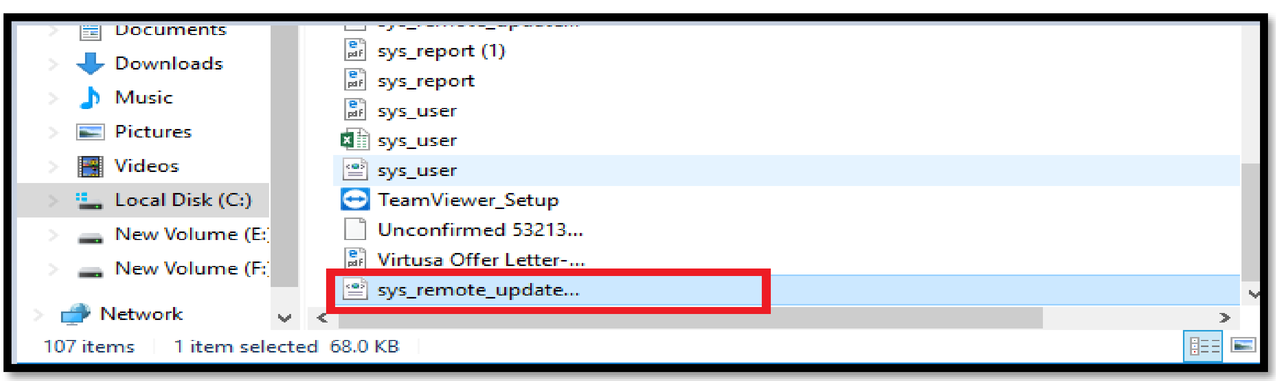
5. Again Open **Same** record



6. Click on **Export to XML (Exporting Changes)**



7. After downloaded **XML file** then take that file and **keep on your desktop**



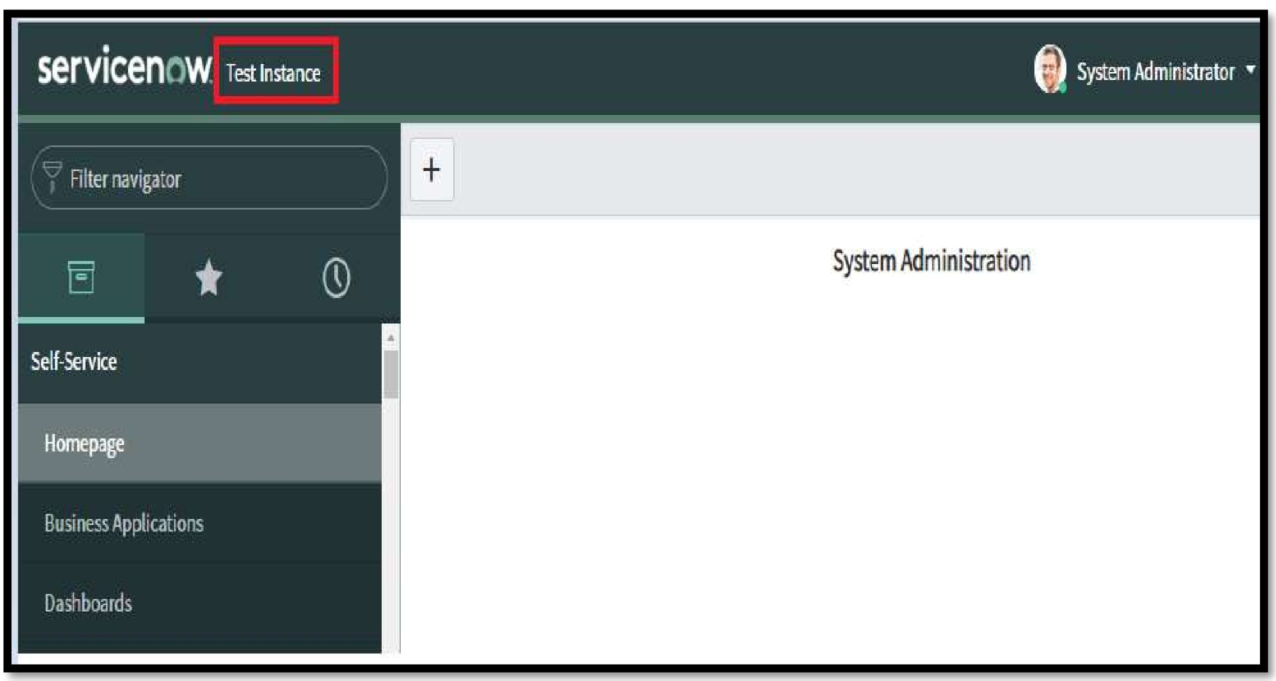
8. Change name of your **Local Update Set** lake below

Exercise :8 Migrating changes to Target Instance (Test)

We are going to move these all changes into testing instance for testing purpose

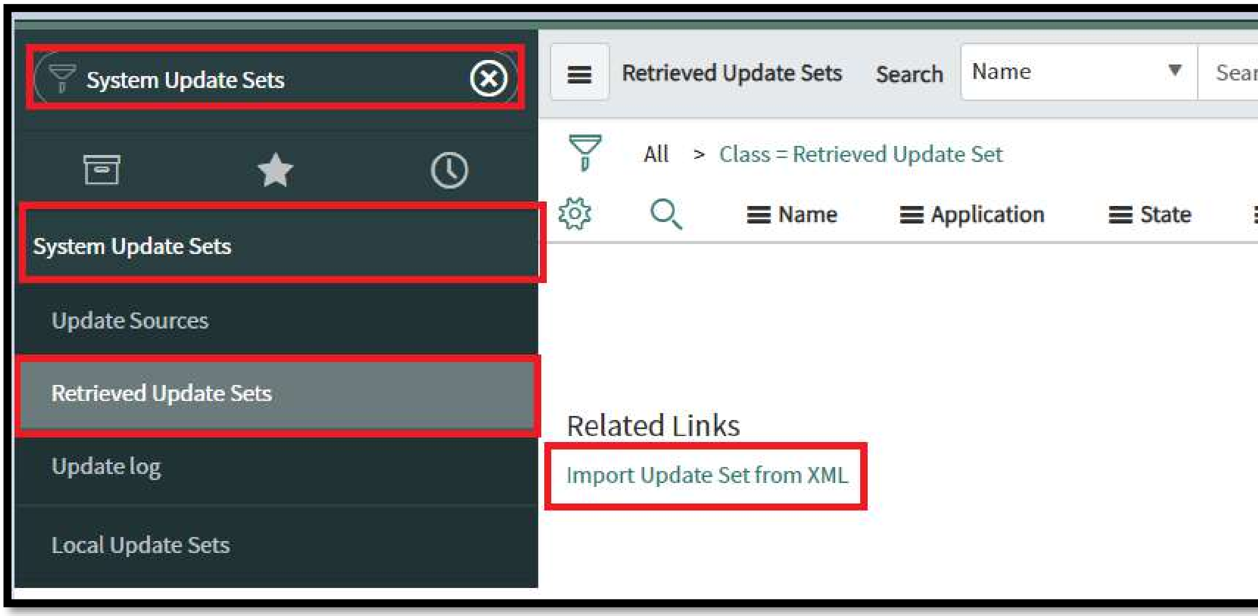
Procedure

1. Open Target Instance **URL >** https://d[redacted].service-now.com/

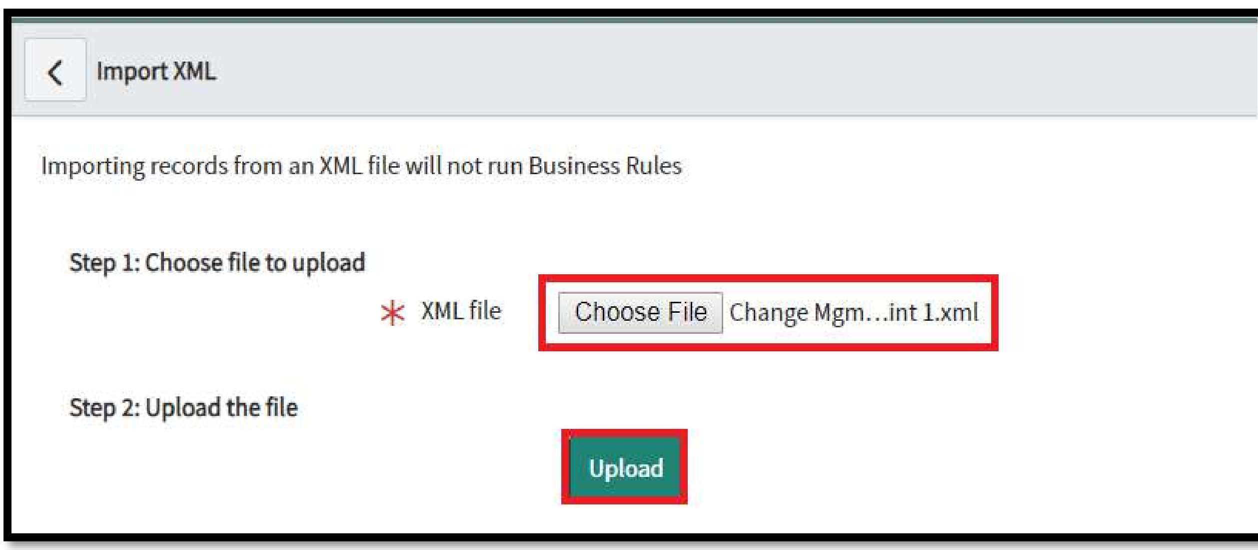


2. Navigate to **System Update Sets > Retrieved Update Sets**

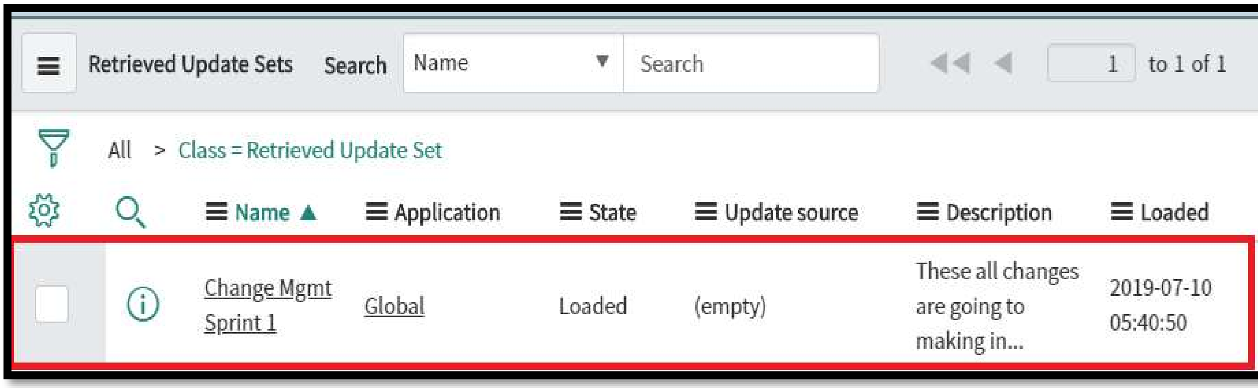
3. Click on **Import Update Set from XML**



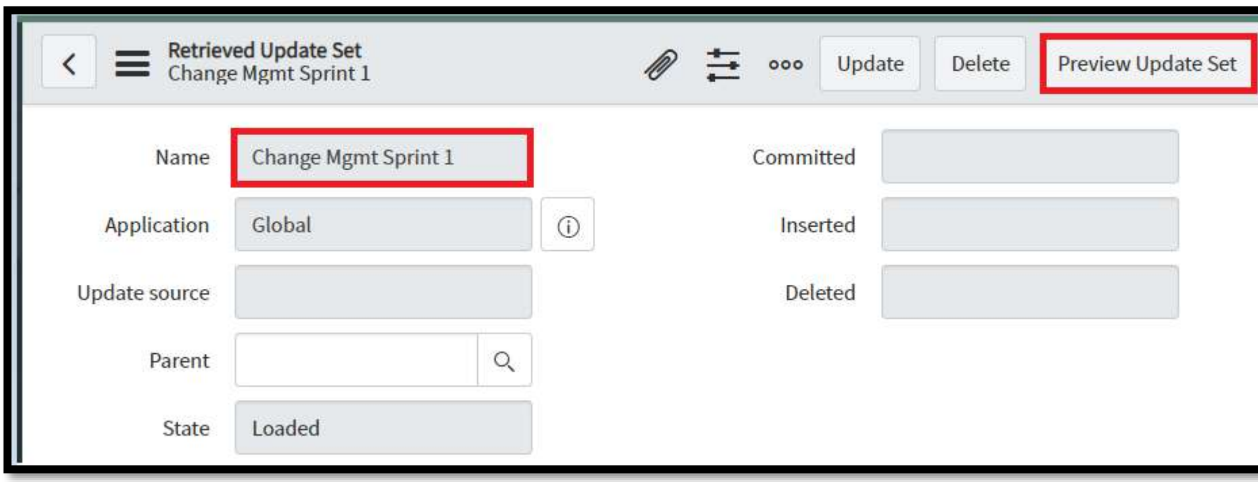
4. Click on **Choose File** > Your exported file
5. Click on **Upload**



6. After imported your file look like below screen shot



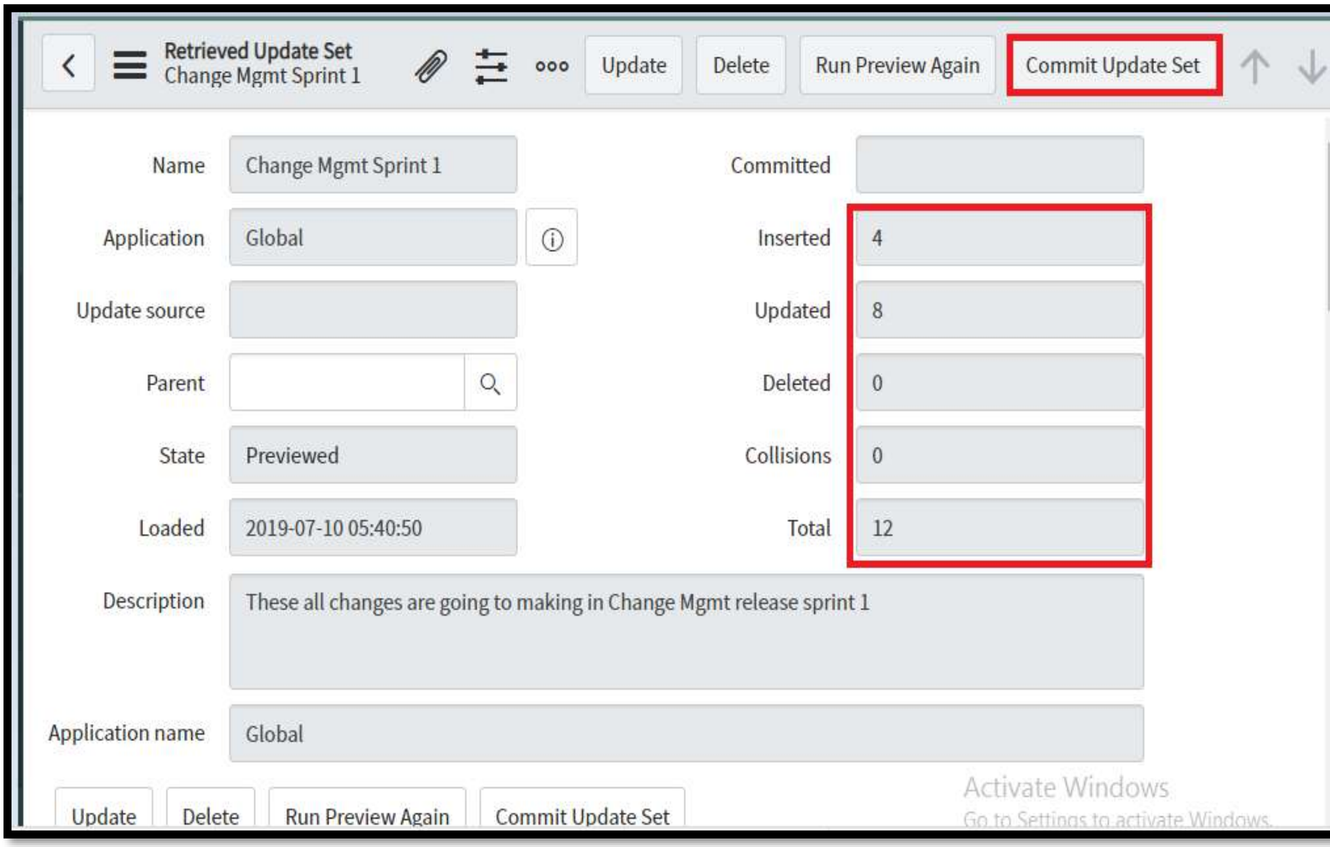
7. Open **Change Mgmt Sprint 1** record
8. Click on **Preview Update Set**



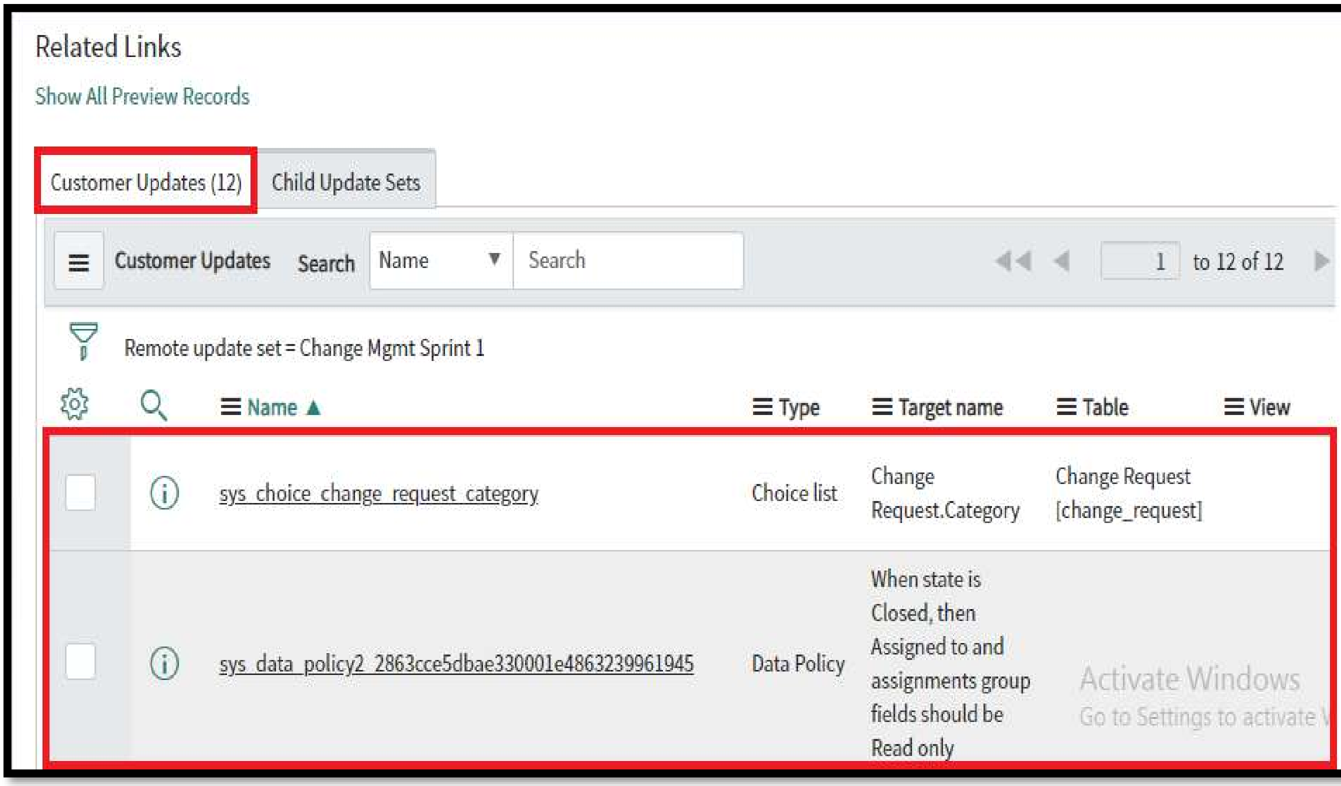
9. We will get below message after clicked on **Preview Update Sets** Button
10. **Succeeded** can define no errors in our XML file
11. Click on **Close**

12. Click on **Commit Update Set**

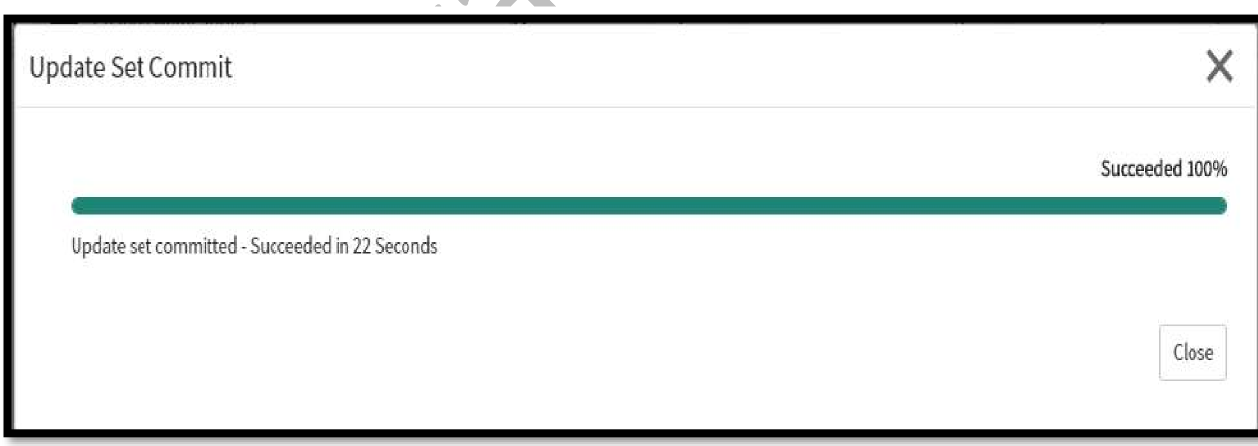
13. **Inserted:** Inserted Changes (4)
14. **Updated:** Updated Changes (8)
15. **Deleted:** No Changes Deleted (0)
16. **Collision:** No Collison (0)
17. **Total:** Total Changes (12)



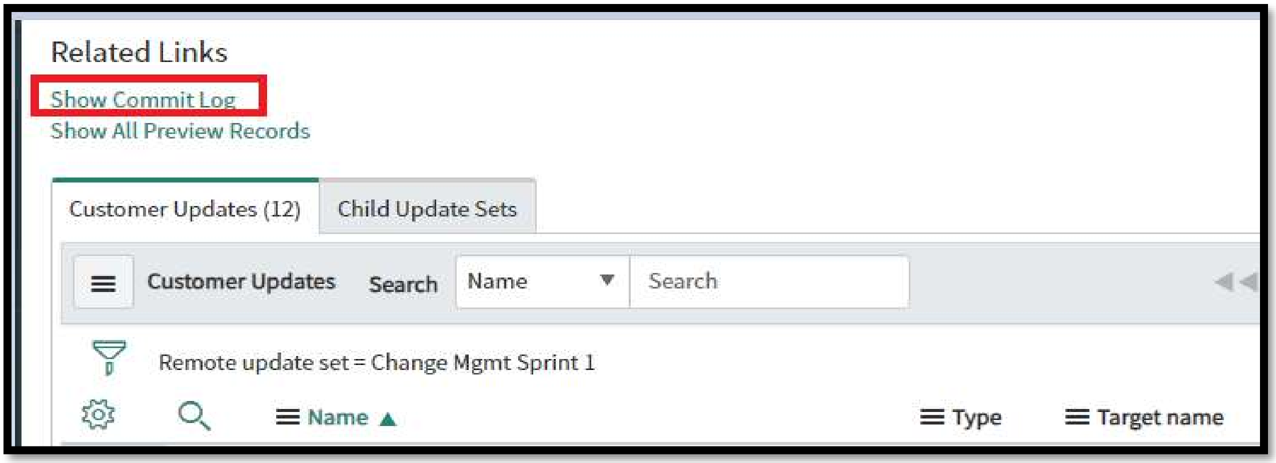
18. Customer Updates



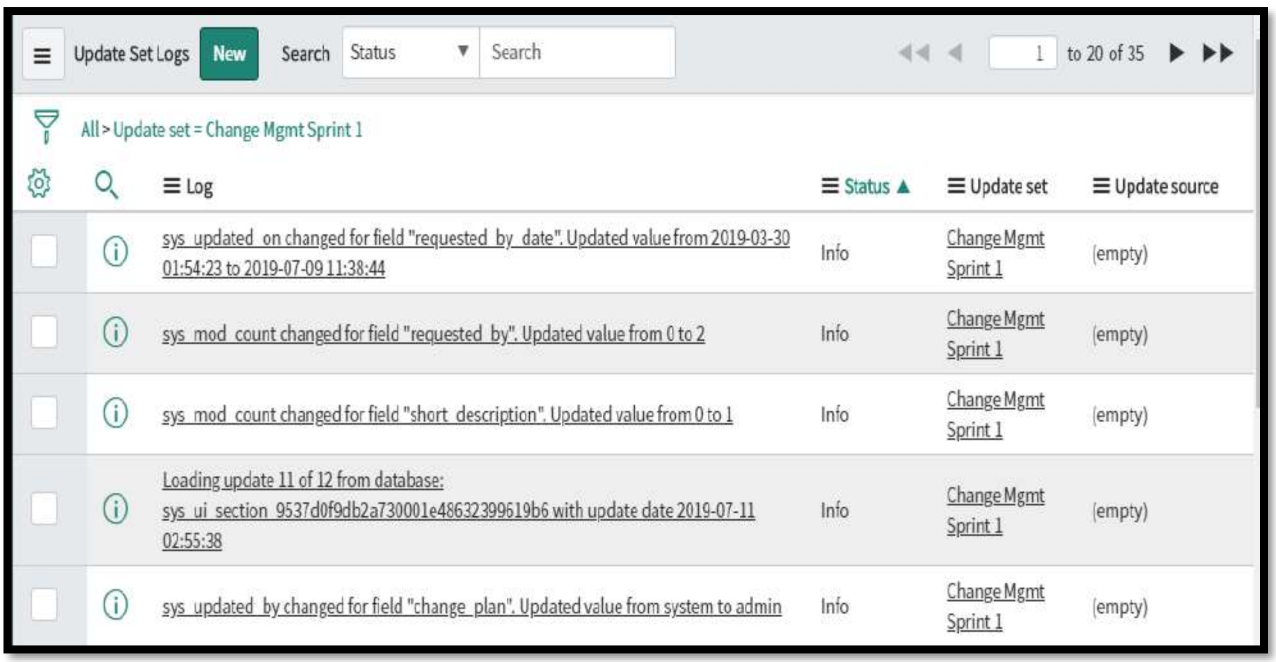
19. After committed changes into target instance get this success message
20. Click on **Close**



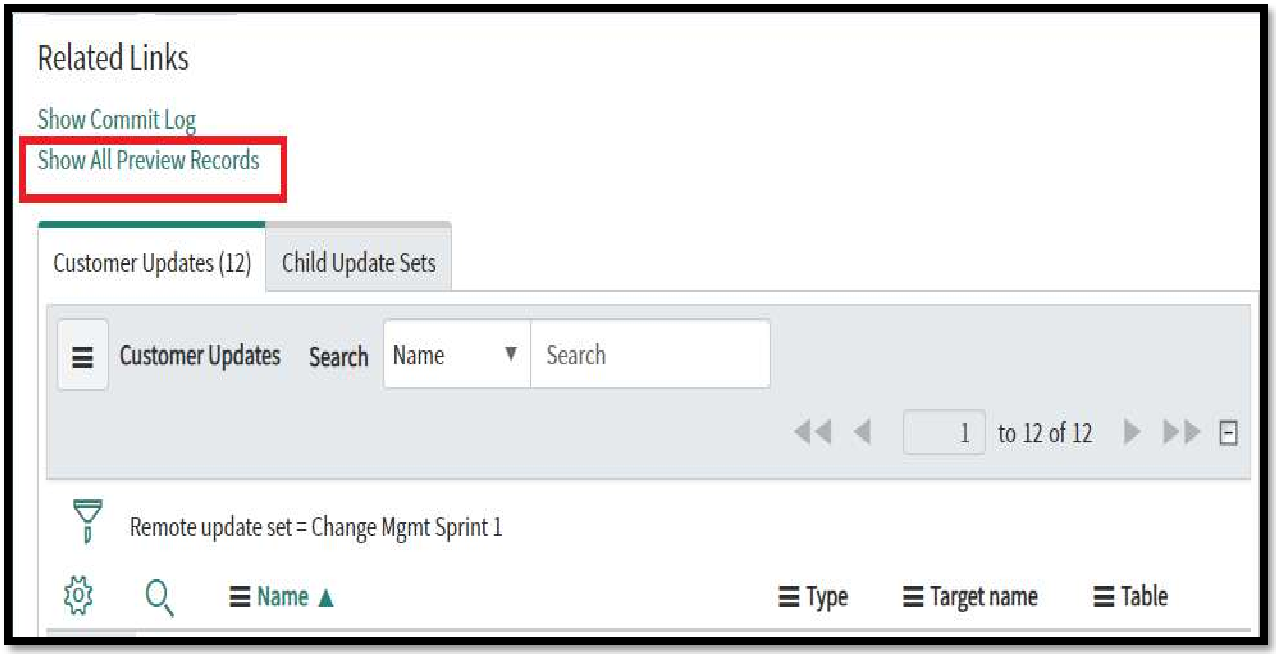
21. Click on **Show Commit Logs** to check update logs



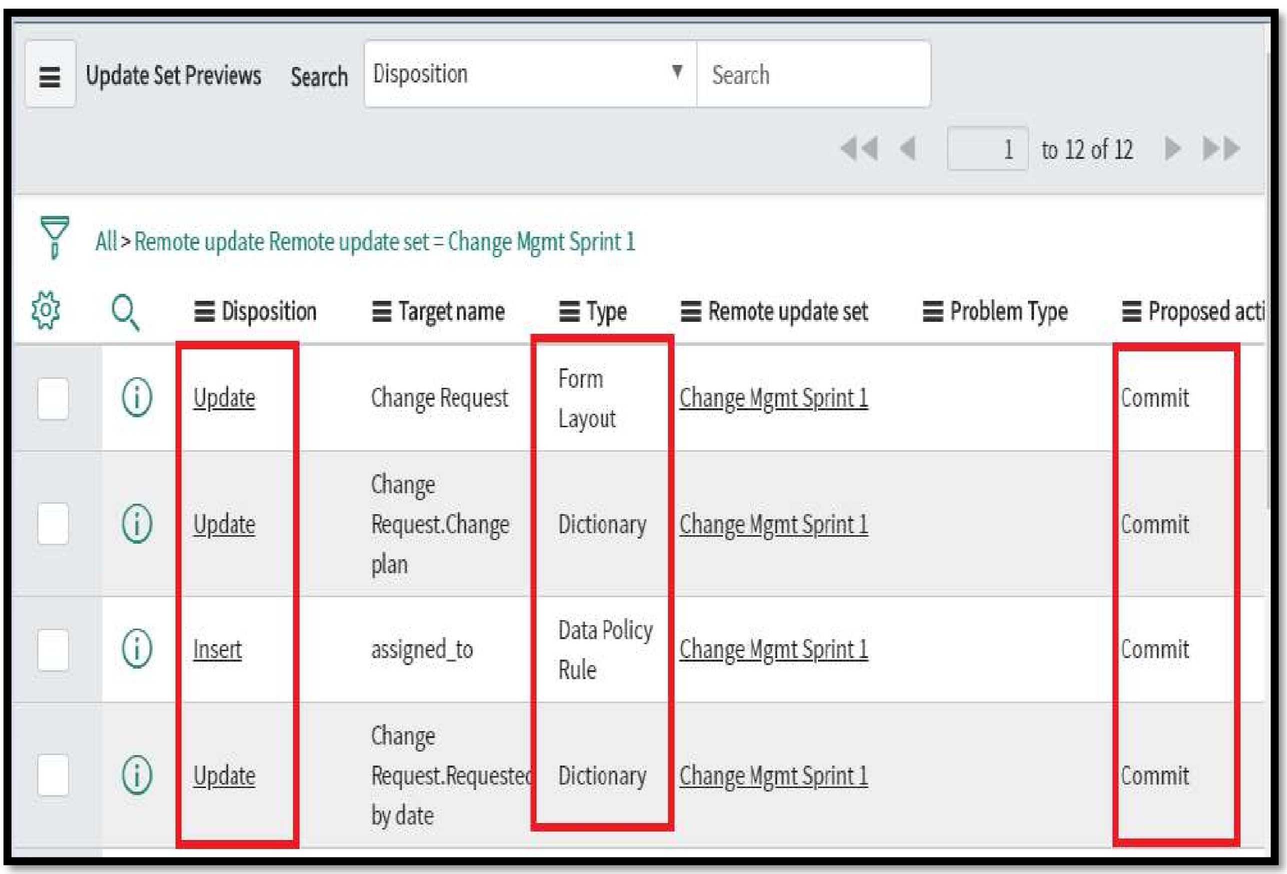
22. Check update logs like below



23. Click on **Show All Preview Record (Inserted or Updated)**



24. Check **update sets previews**



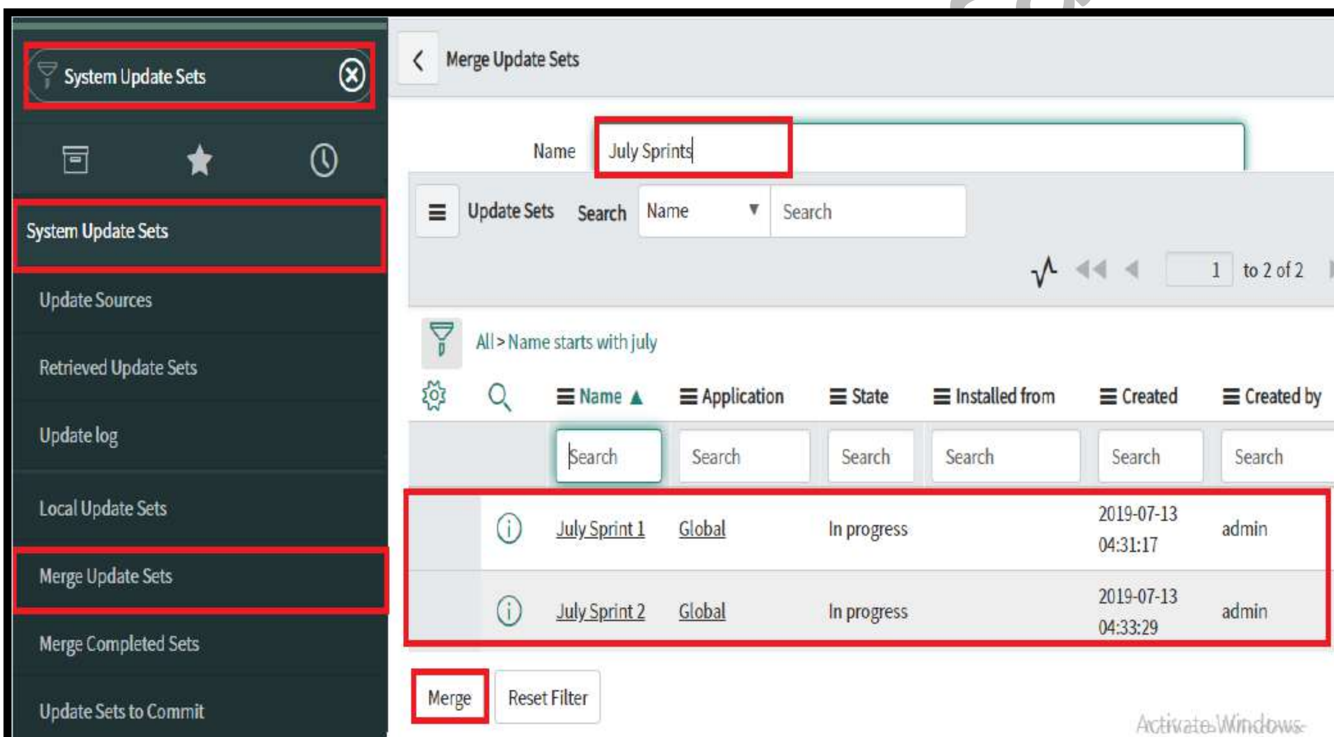
Working with Merge Update Sets

We can merge multiple update sets into a single update set. The newer batch update sets feature accomplishes the same outcome with a more predictable and fast solution moving changes from to another instance.

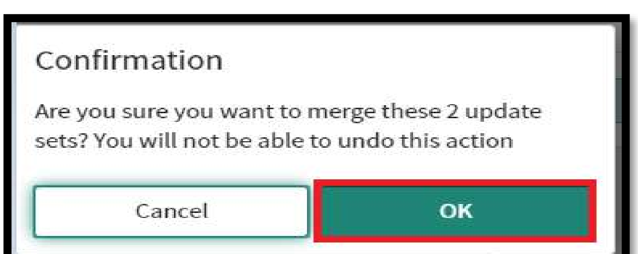
Note: You cannot **"unmerge"** update sets once they have been merged.

Procedure

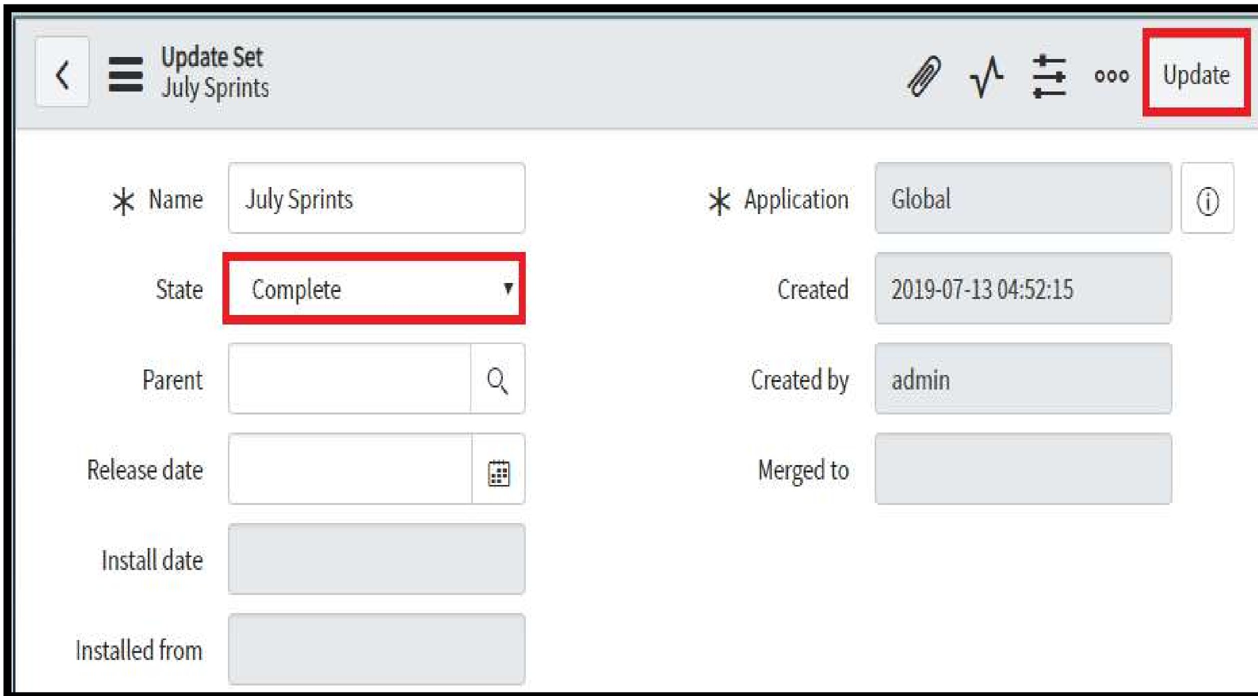
1. Navigate to **System Update Sets > Merge Update Sets**.



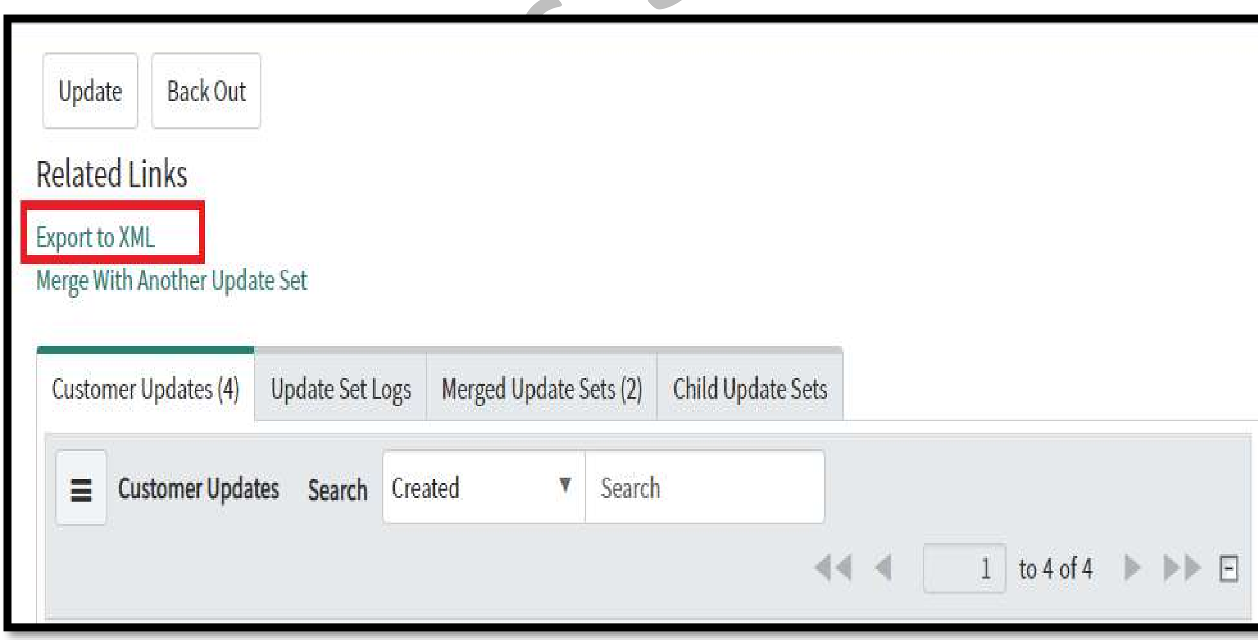
2. **Name:** July Sprints
3. Select your update sets to merging
4. Click on **Merge**
5. Click on **Save**



6. Again open **Merged Sets** record



7. Change State to **Complete**
8. Click on **Update**



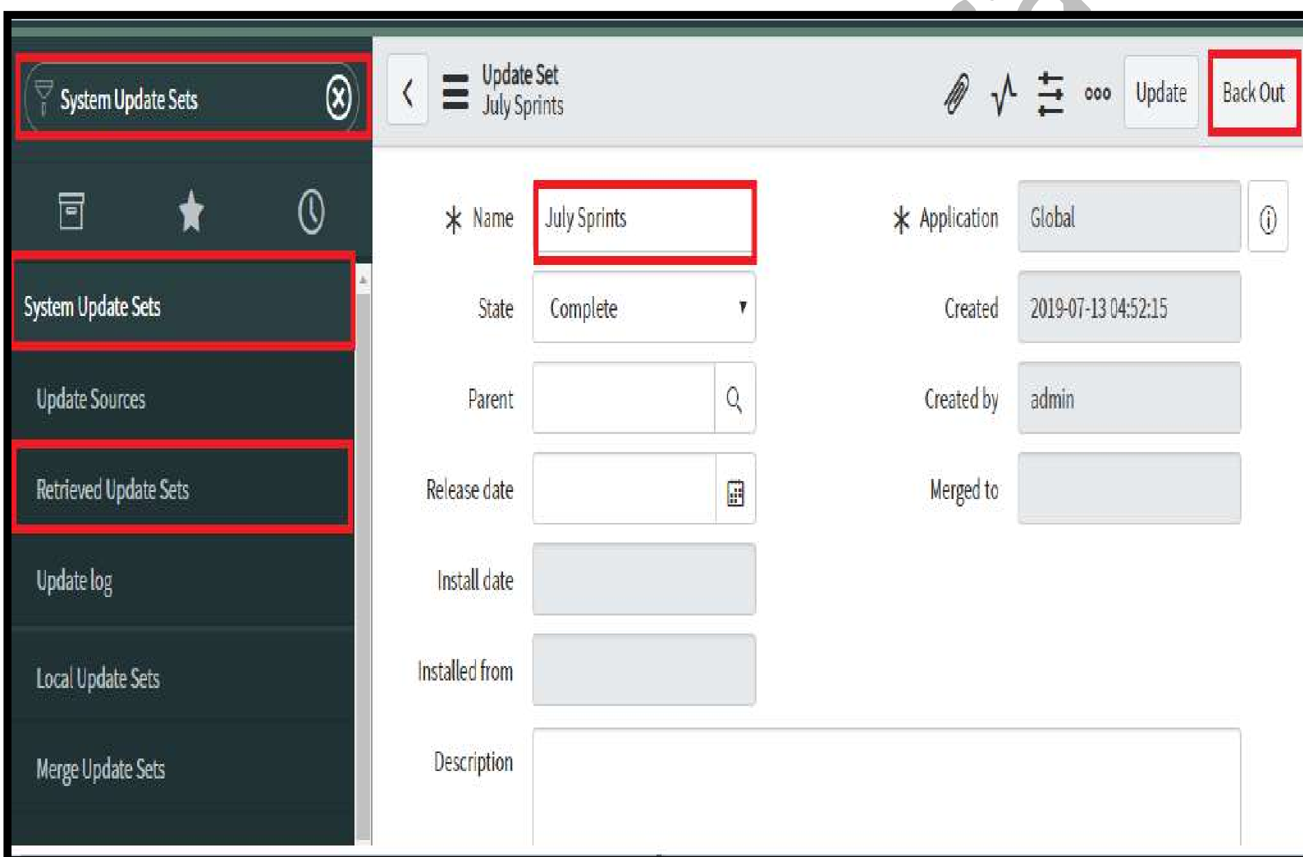
9. Again open same record
10. Click on **Export to XML**
11. Once exported XML file then the file can move to target instance as followed above steps

Back out changes from an Update Set

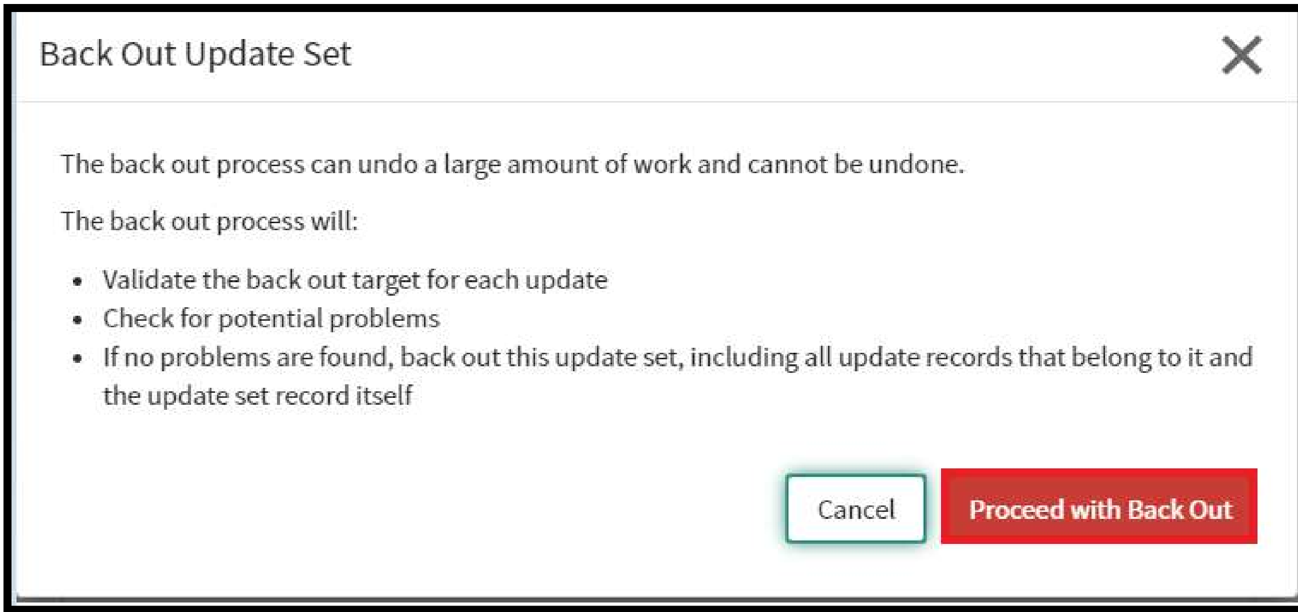
We can be able to back out changes to existing records for any committed update set. If the changes are not working

Procedure

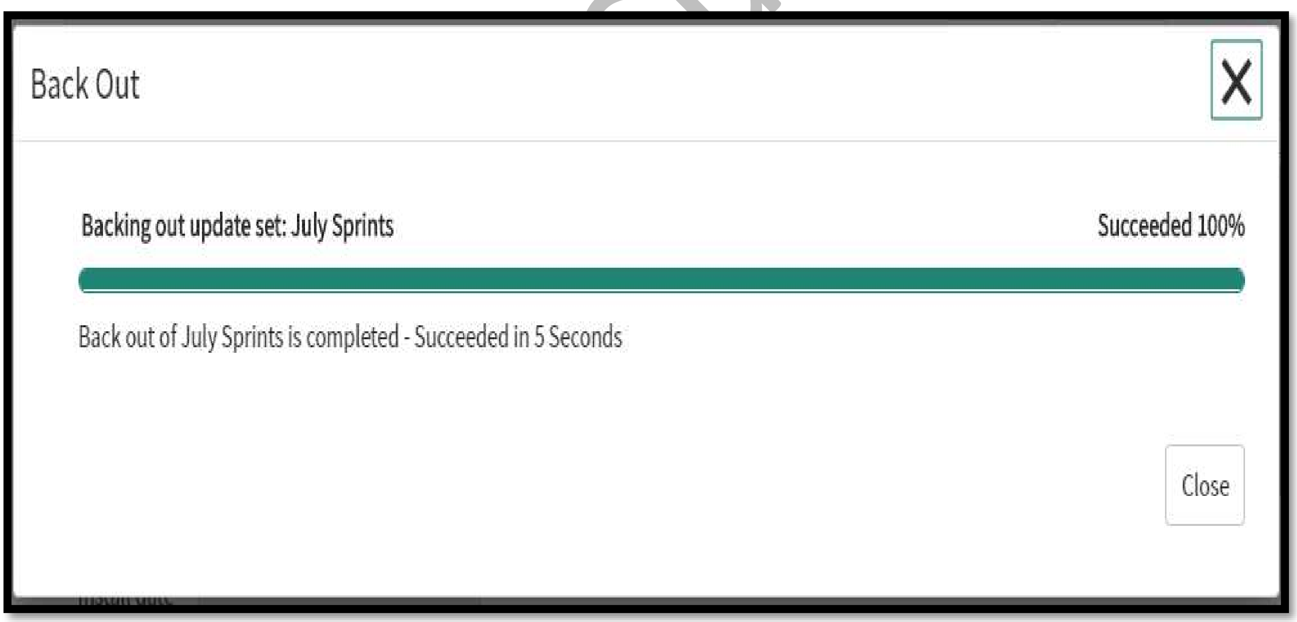
1. Navigate to **System Update Sets > Retrieved Update Sets** or **System Update Sets > Local Update Sets.**
2. Open the **Update Set** record.



3. Click on **Back Out**



4. Click on **Proceed with Back Out**
5. Successfully **Black Out** the changes from expected instance



6. Click on **Close**
7. Whether successfully **back Out or not**

Taking some precautions working with Update Sets

Because Update Sets make changes to an instance, review the following best practice information to avoid errors and performance issues.

1. Do not delete Update Sets. If an Update Set is deleted, any updated records may be overwritten in the next update.
2. Do not include the **system_id** field from the **ldap_server_config** record in an Update Set. An Update Set from a working configuration points to the wrong system_id node for the target instance and does not work.
3. Do not back out the Default Update Set. This action causes damage to the system.
4. Never change the Update Set field value **(update_set)** in a Customer Update record **(sys_update_xml**). If a customization is made in the wrong Update Set, take the following action:
5. Switch to the desired **Update Set.**
6. Modify the object (record) that was originally changed. You can make a trivial change, such as adding a field.
7. **Save** the record
8. Back out the change just performed, and then save the record again.
9. Always preview an Update Set before committing it.
10. Set completed Update Sets on the production instance to **ignore**. This state ensures the Update Set is not reapplied when cloning the instance.
11. Keep a to-do list of manual changes and data loads that need to be completed after an Update Set is applied.
12. Do not make too many changes at one time. Verify that the correct changes have been made incrementally.

Chapter – 16 Service Catalog

What is Service Catalog?

1. **Service Catalog** is an application with in platform
2. Create service catalogs that provide your customers with **self-service** opportunities.
3. Customize portals where your customers can request catalog items such as service and product offerings.
4. The Service Catalogue is the only part of the ITIL Service Portfolio published to Customers,
5. It is used to support the **Sale and Delivery** of IT Services.
6. The Service Catalogue includes information about **deliverables, prices, contact points, ordering and request Processes.**
7. With the Service Catalog application, you can create one or more multiple catalogs of products and shared services
8. Catalog items are goods or services available to order from the service catalog. Items can be anything from **Software, Hardware, like Tablets and Phones, Software Applications, Furniture and Office Supplies.**

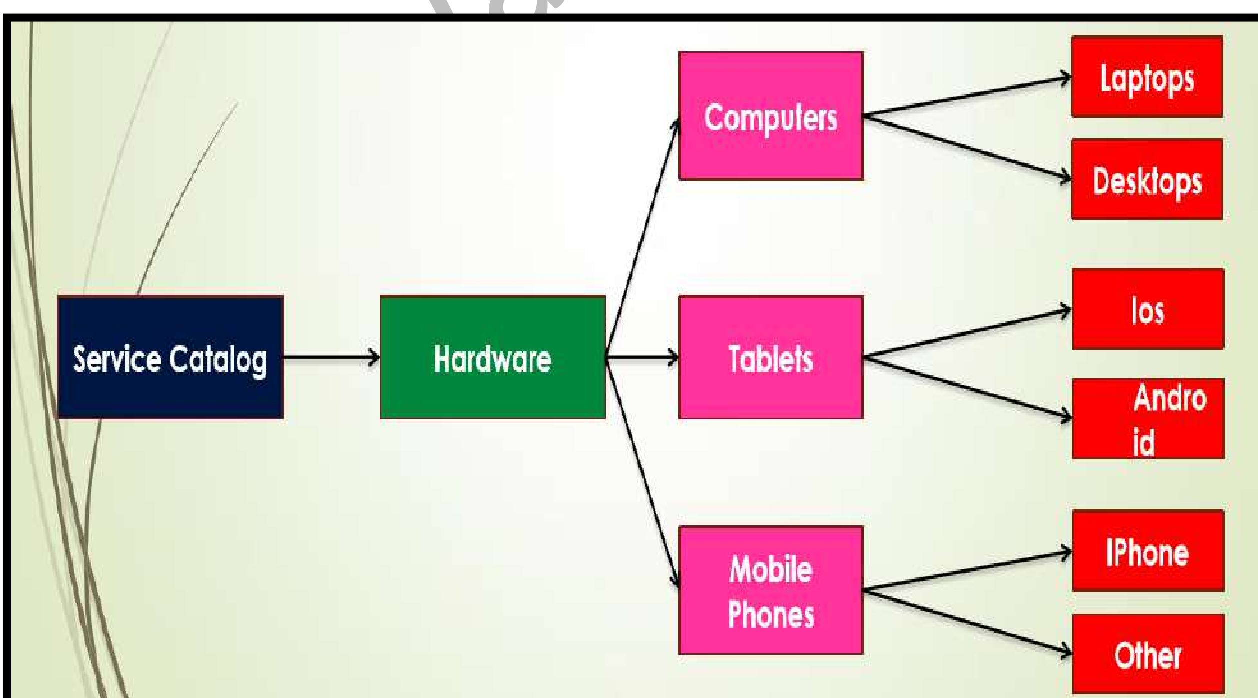
Service Catalog offers to

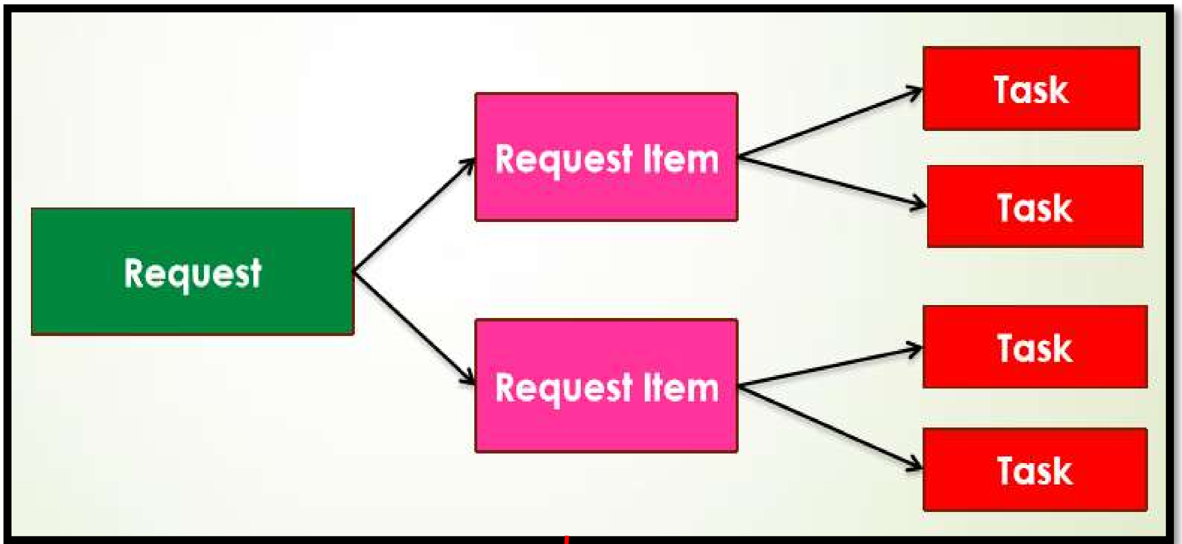
Service Catalog contain almost anything

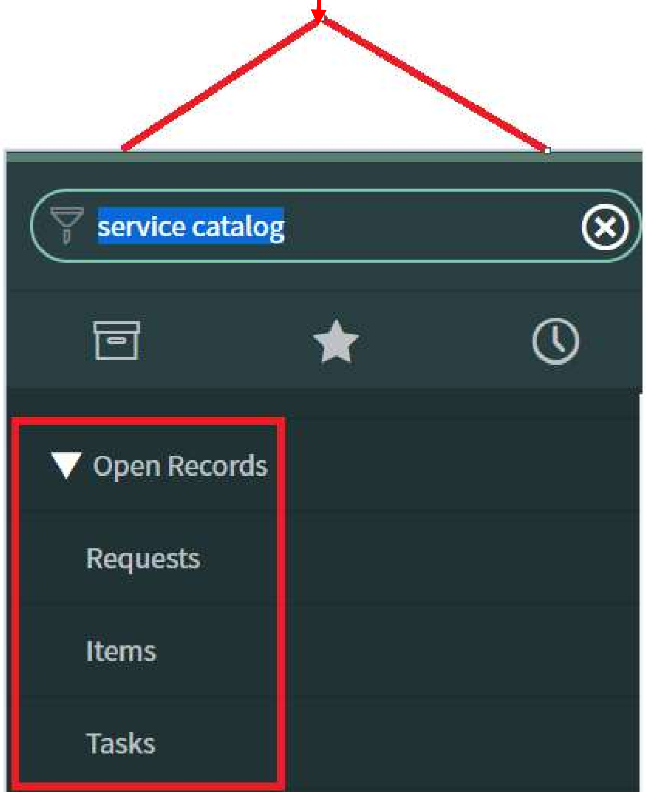
- ✓ Products like computer equipment,
- ✓ Software,
- ✓ Furniture
- ✓ Office supplies
- ✓ Service like work requests,
- ✓ Password reset
- ✓ Support to resolve incidents
- ✓ Record producers
- ✓ Order guides
- ✓ Workflows

Service Catalog Architecture

1. The products and services in a catalog are organized into categories and sub categories
2. The base system includes several useful categories, and you can define your own




Request
Request Item
Request Item
Task
Task
Task
Task


service catalog
Open Records
Requests
Items
Tasks

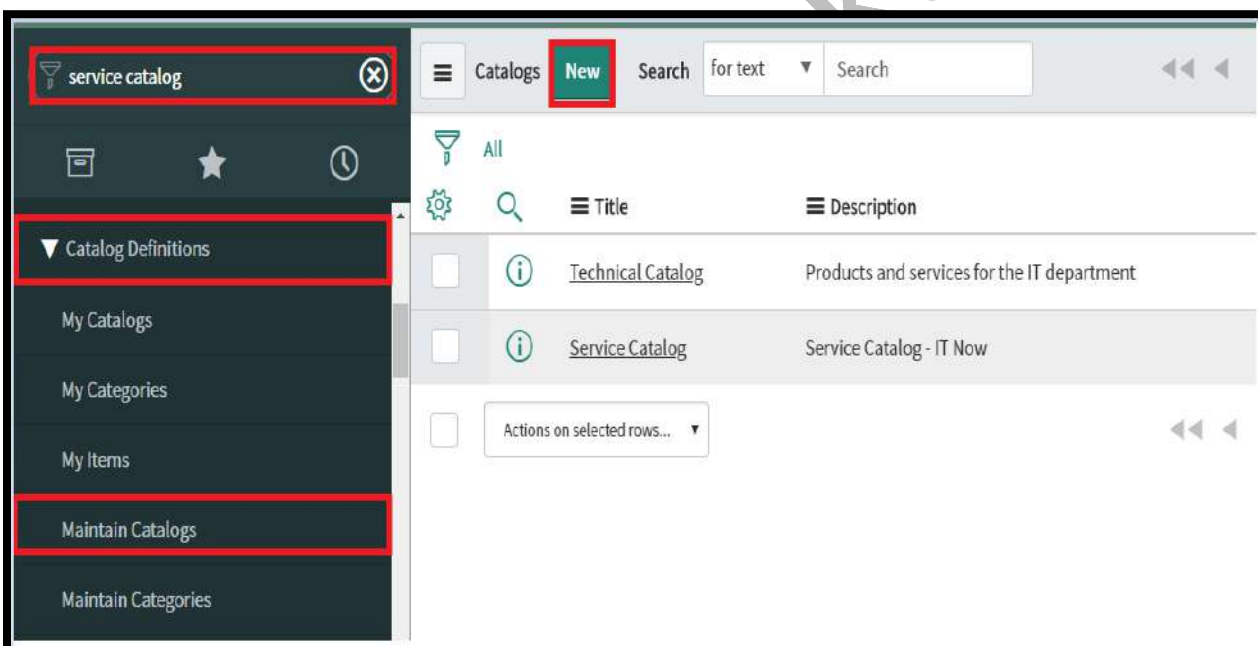
Exercise: 1 Configure Service Catalog

Service Catalog provide the opportunities to users with the Catalog Administrators role **(catalog_admin)** to configure service catalog with in instance

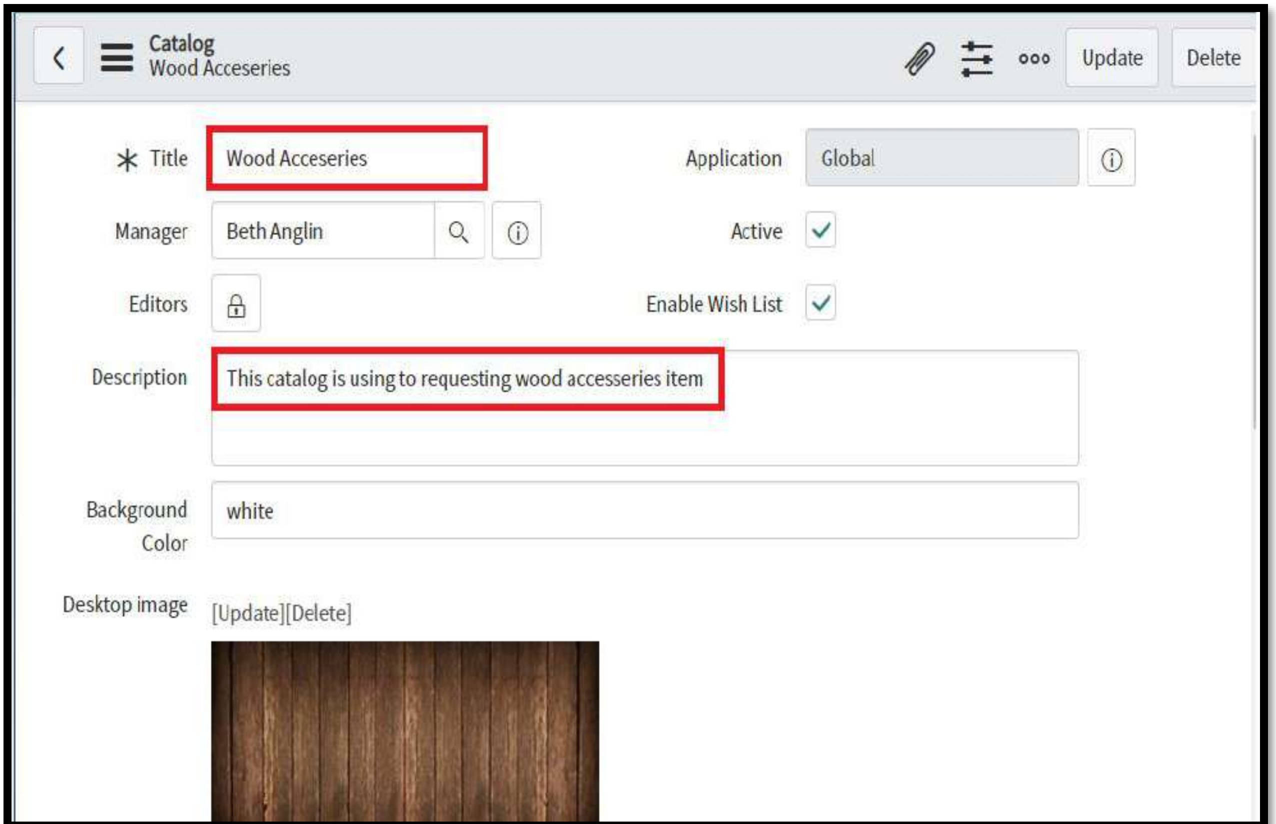
Service Catalog Application will help us to configure one or more service catalogs and **provides self-service opportunities** and better user interface **(Self-Service)**. We can organize our catalog offerings by using the ability to publish multiple catalogs targeted to specific users.

Procedure

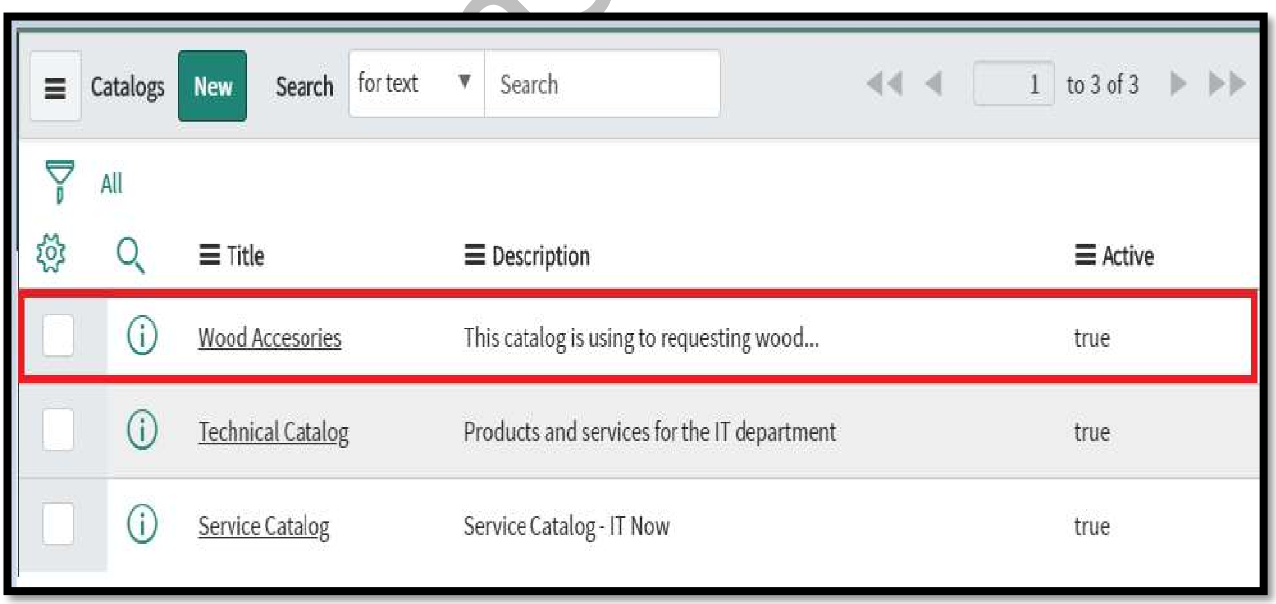
1. Navigate to **Service Catalog > Catalog Definition > Maintain Catalog**
2. Click on **New**
3. Fill the **Service Catalog** Form



4. **Title:** Wood Accessories
5. **Manager:** Select Manager (A catalog manager can define and maintain a single catalog)
6. **Enable Wish List:** Check **True**
7. **Description:** This catalog is using to requesting wood accessories item
8. **Background Colour:** Default **(White)**
9. **Active:** True (This Catalog **Visible** to all user)
10. **Desktop Image:** Add your desktop mage **(Based on our requirement)**
11. **Editors: Empty (**Who can edit and modified this catalog)
12. Click on **Submit**



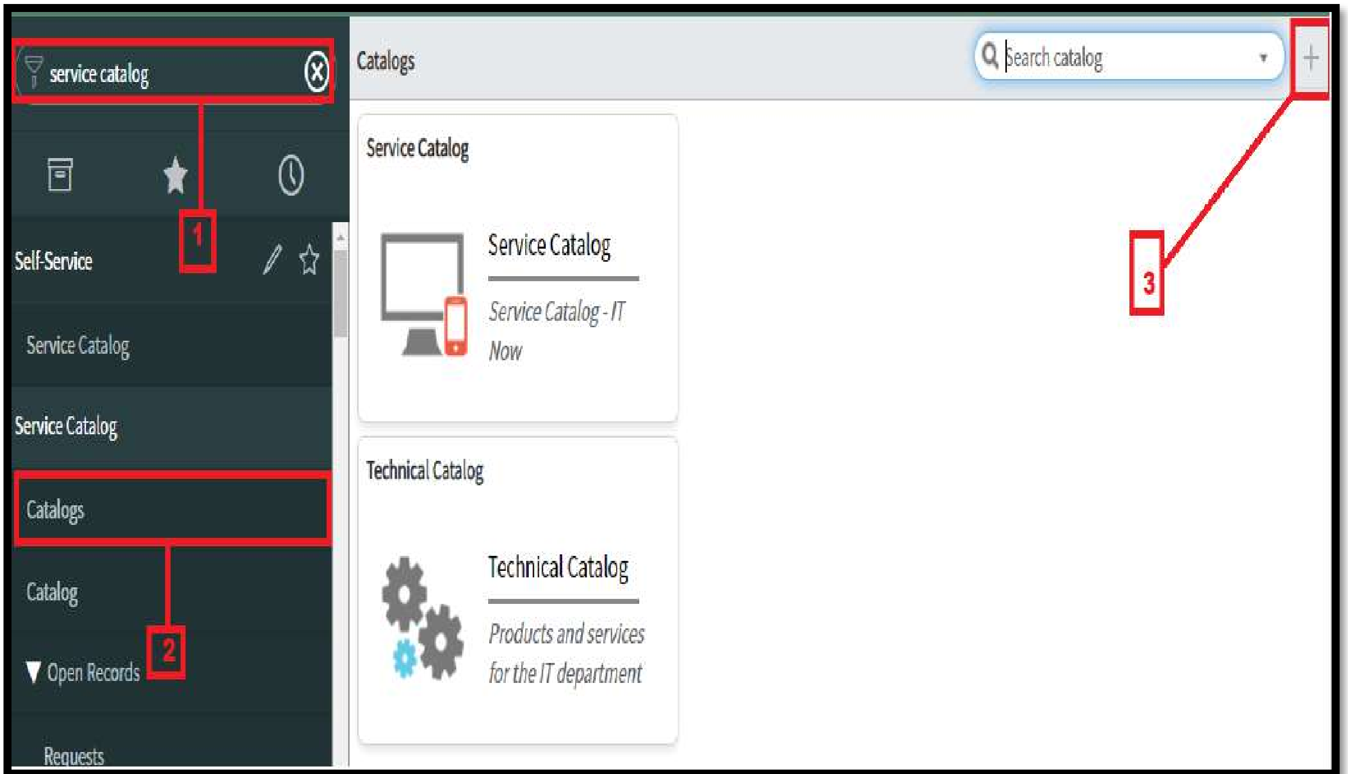
13. Successfully configured new **Catalog** into list of Catalog



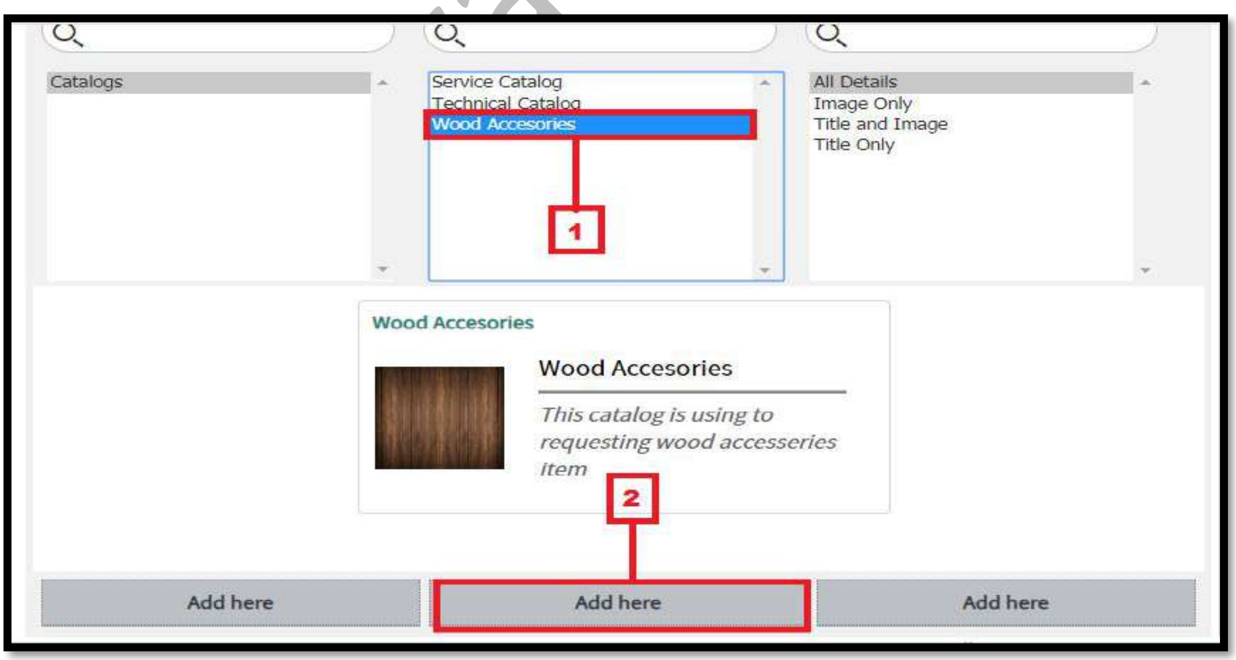
14. Add this new catalog to catalog **Home Page**

15. Navigate to **Service Catalog > Catalogs**

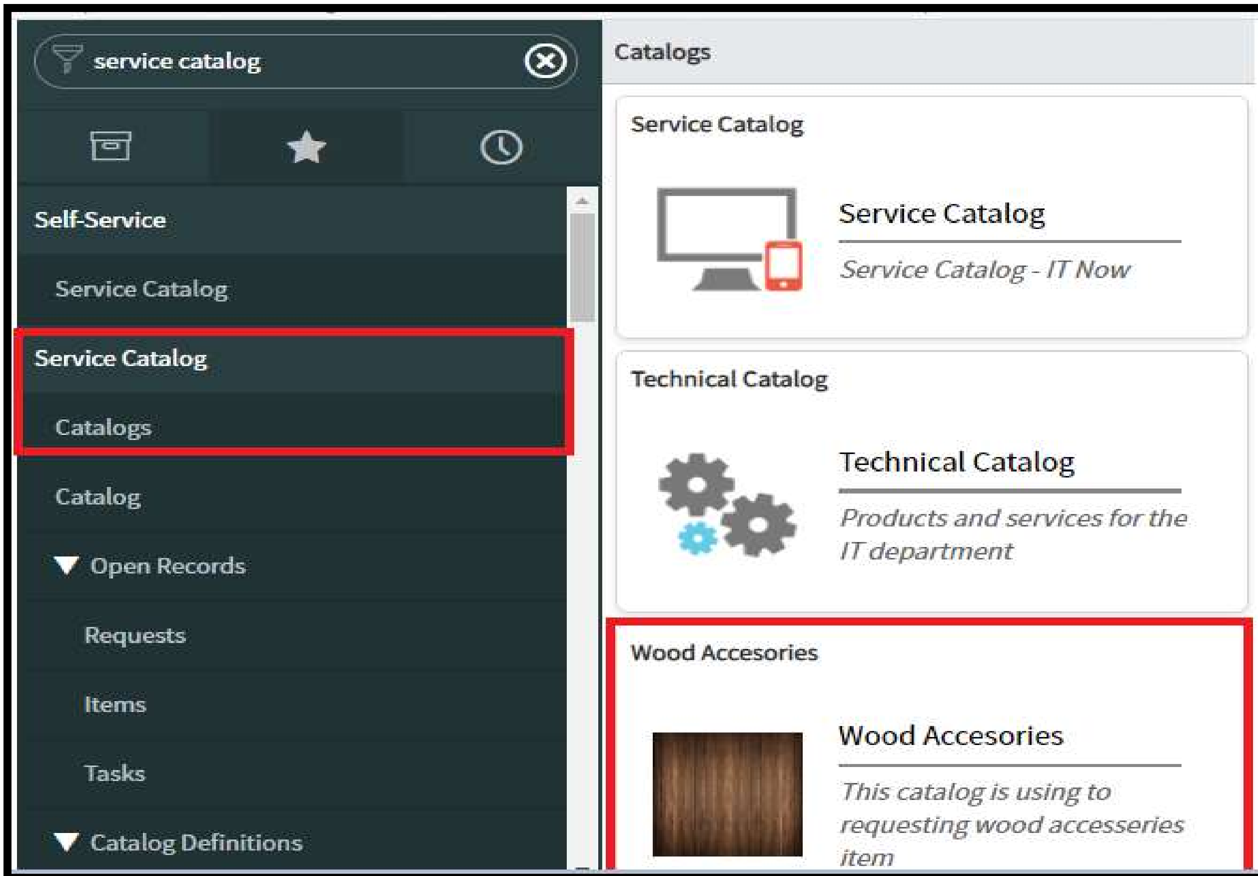
16. Click on **+ icon**



17. Select **Wood Accessories** Service Catalog
18. Click on **Add Here**
19. Click on **Close** Icon



20. Successfully added new **Service Catalog** (Like - Below screenshot)



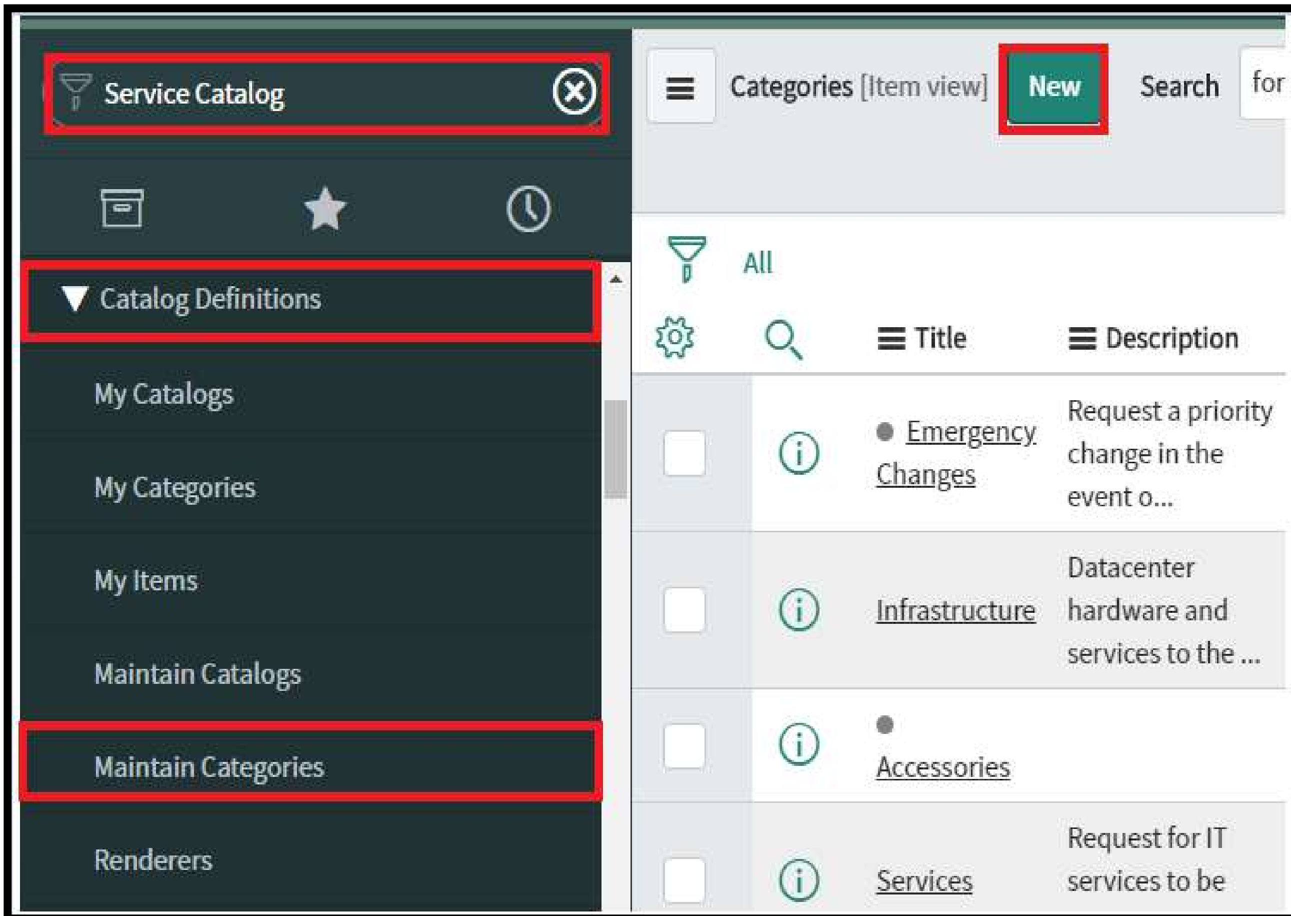
Exercise: 2 Configure Service Catalog Category (Wood Chair)

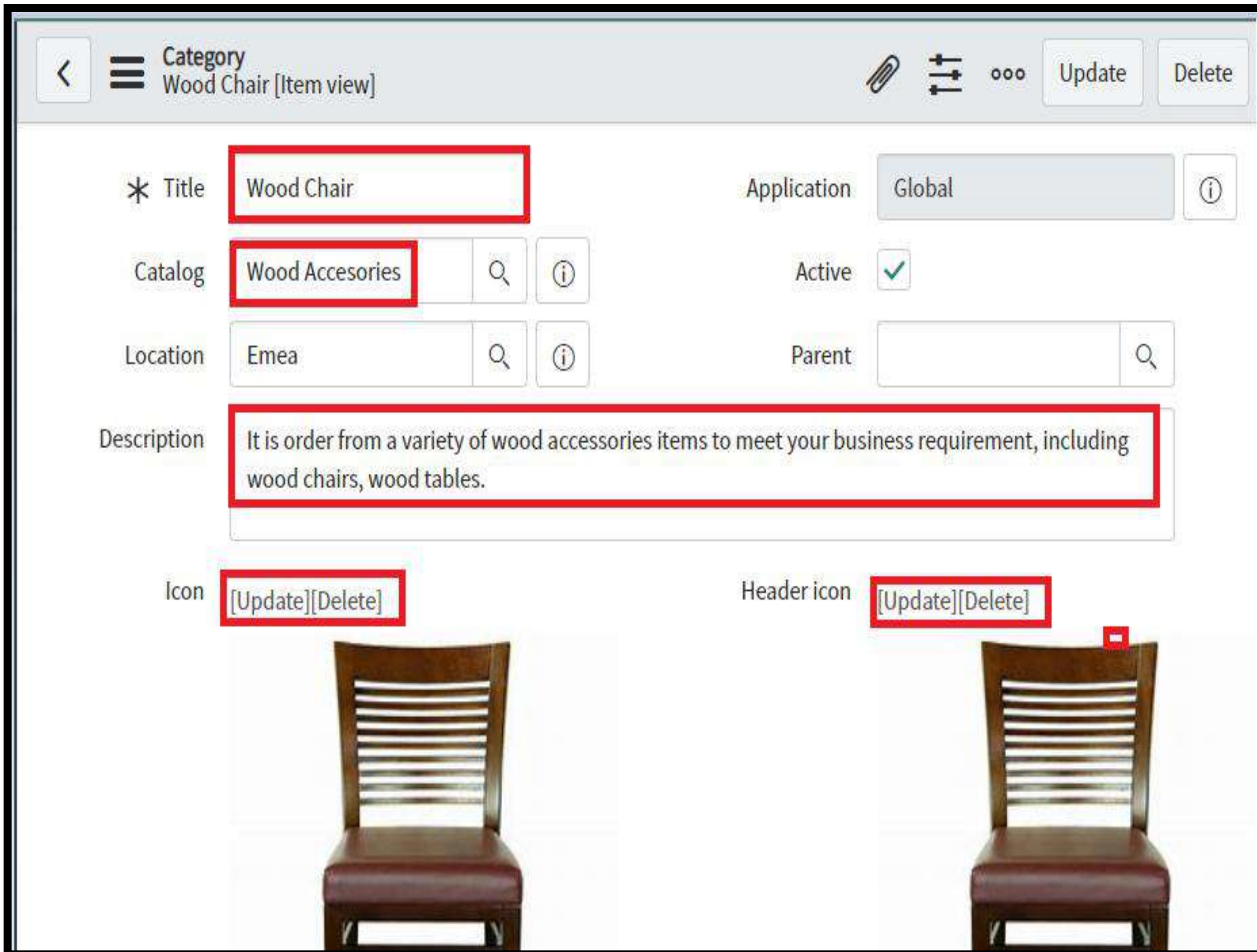
Service Catalog Categories can manage service catalog items into different sub groups. Administrators and catalog administrators can create and configure categories, defining their characteristics and adding content such as catalog items to them based on our client requiremt

Categories may contain a parent-child relationship, for example, **Finance** and **IT** and **Laptops**. A child category is a subcategory of its parent category. Each **Catalog Item, Order Guide, Record Producer**, **Content Item**, and subcategory appears as a single item within the category. A catalog item can exist in multiple categories, giving you the flexibility to offer the same service from multiple places within your catalog.

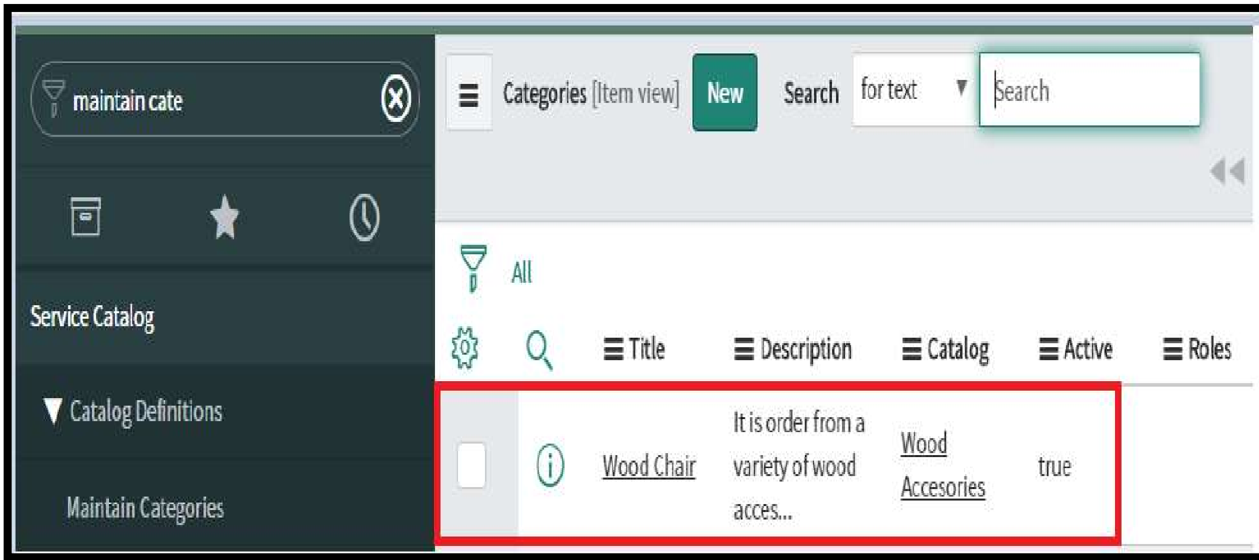
Procedure

1. Navigate to **Service Catalog > Catalog Definition > Maintain Categories**
2. Click on **New**
3. Fill the **Service Catalog Categories** Form

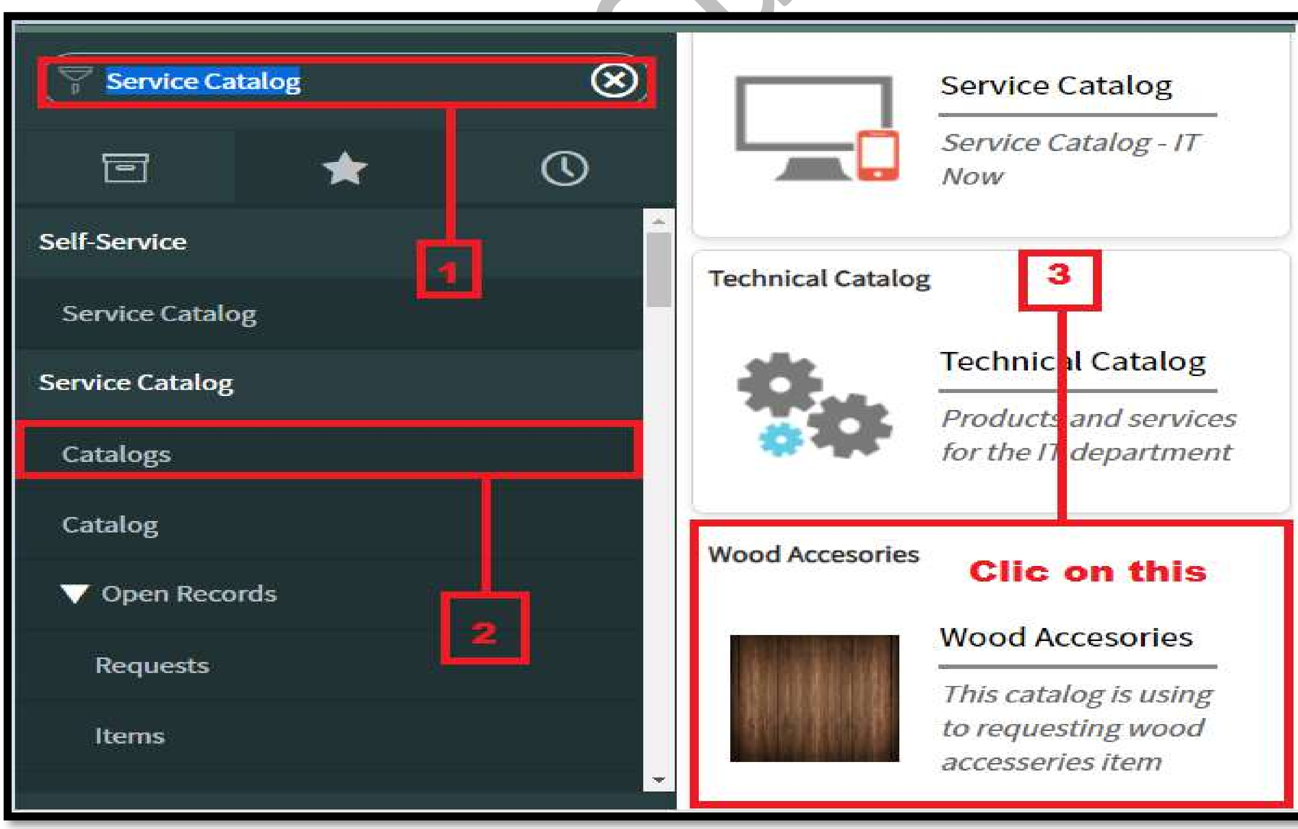




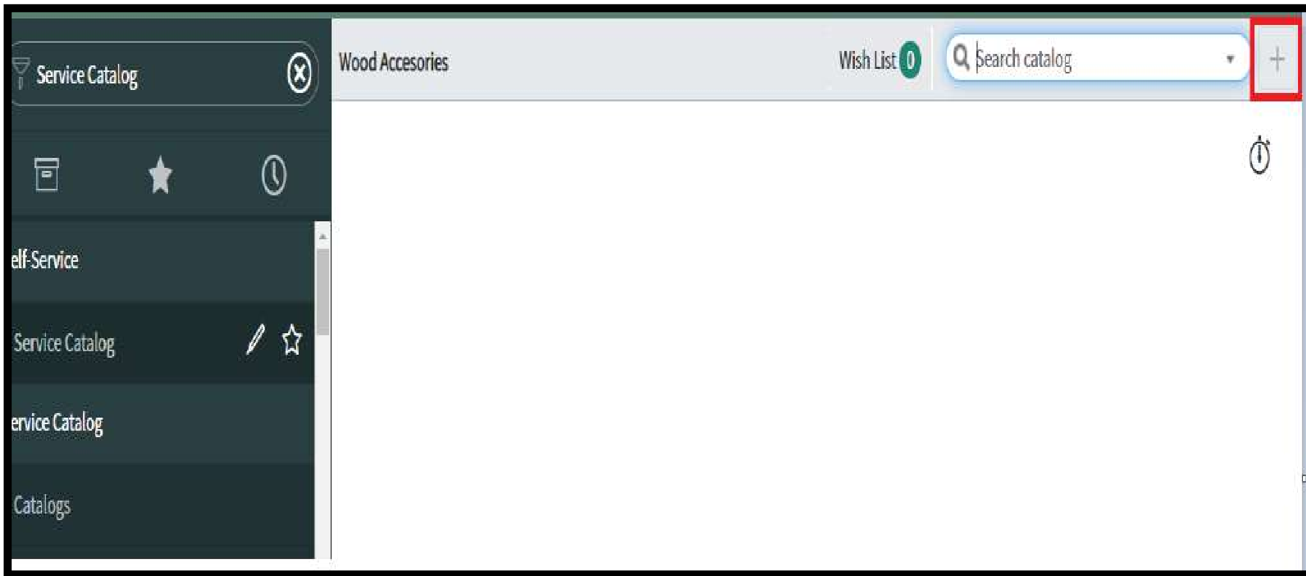
4. **Title:** Wood Chair
5. **Catalog:** Wood Accessories
6. **Location:** Optional
7. **Description:** It is order from a variety of wood accessories items to meet your business requirement, including **wood chairs, wood tables.**
8. **Icon:** Add
9. **Header Icon:** Add
10. Click on **Submit**



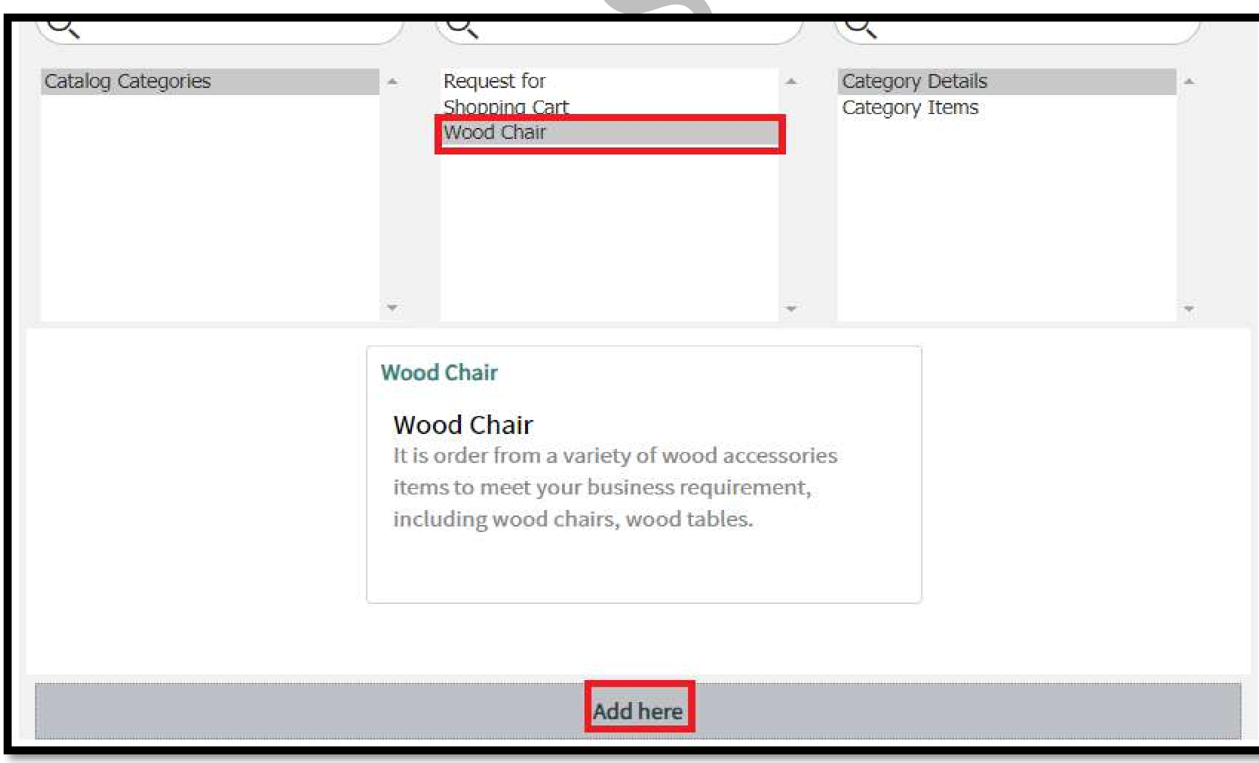
11. Successfully configured new **Category** into list of list of **Categories**
12. Add this new **Category** to category **Home Page**
13. Navigate to **Service Catalog > Catalogs > Wood Accessories**



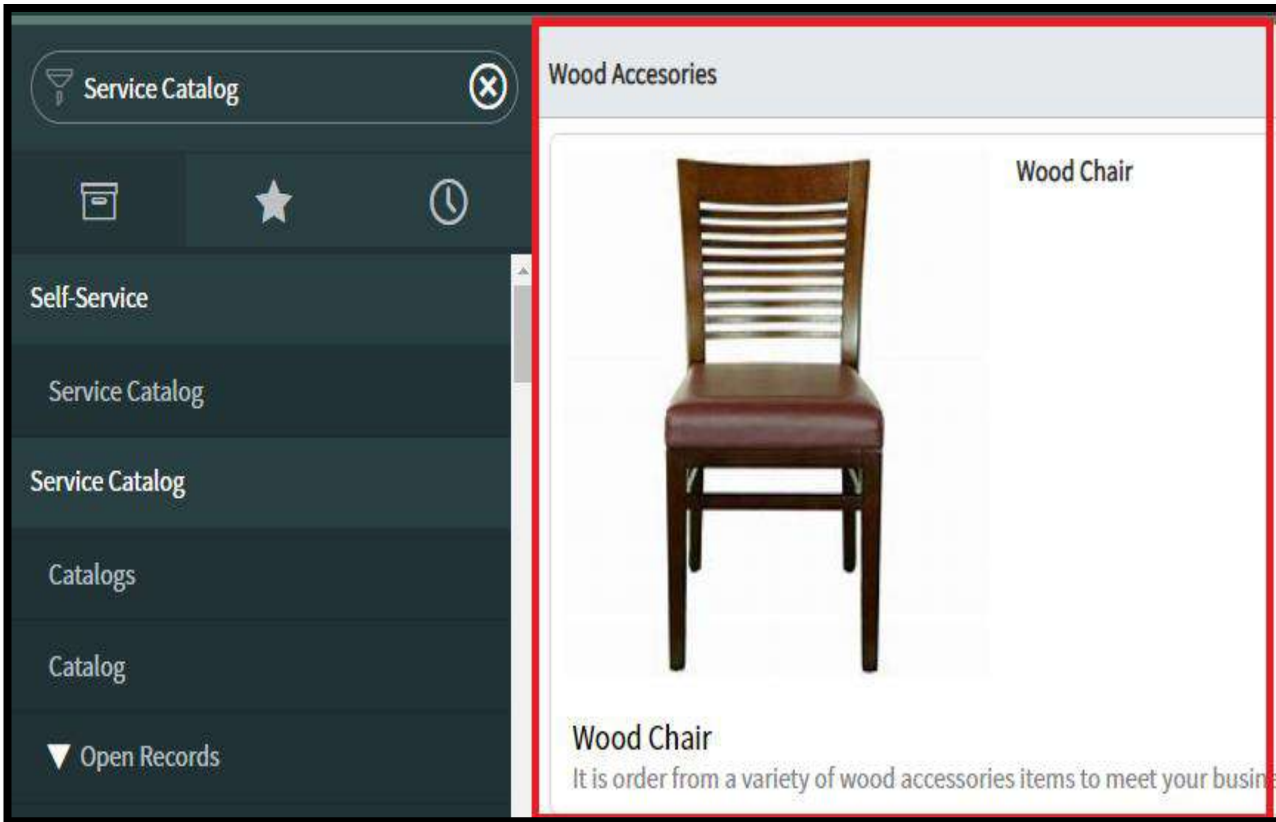
14. Click on **+ icon**



15. Select **Wood Chair**
16. Click on **Add here**



17. Successfully added new **Service Category** (Like - Below screenshot)



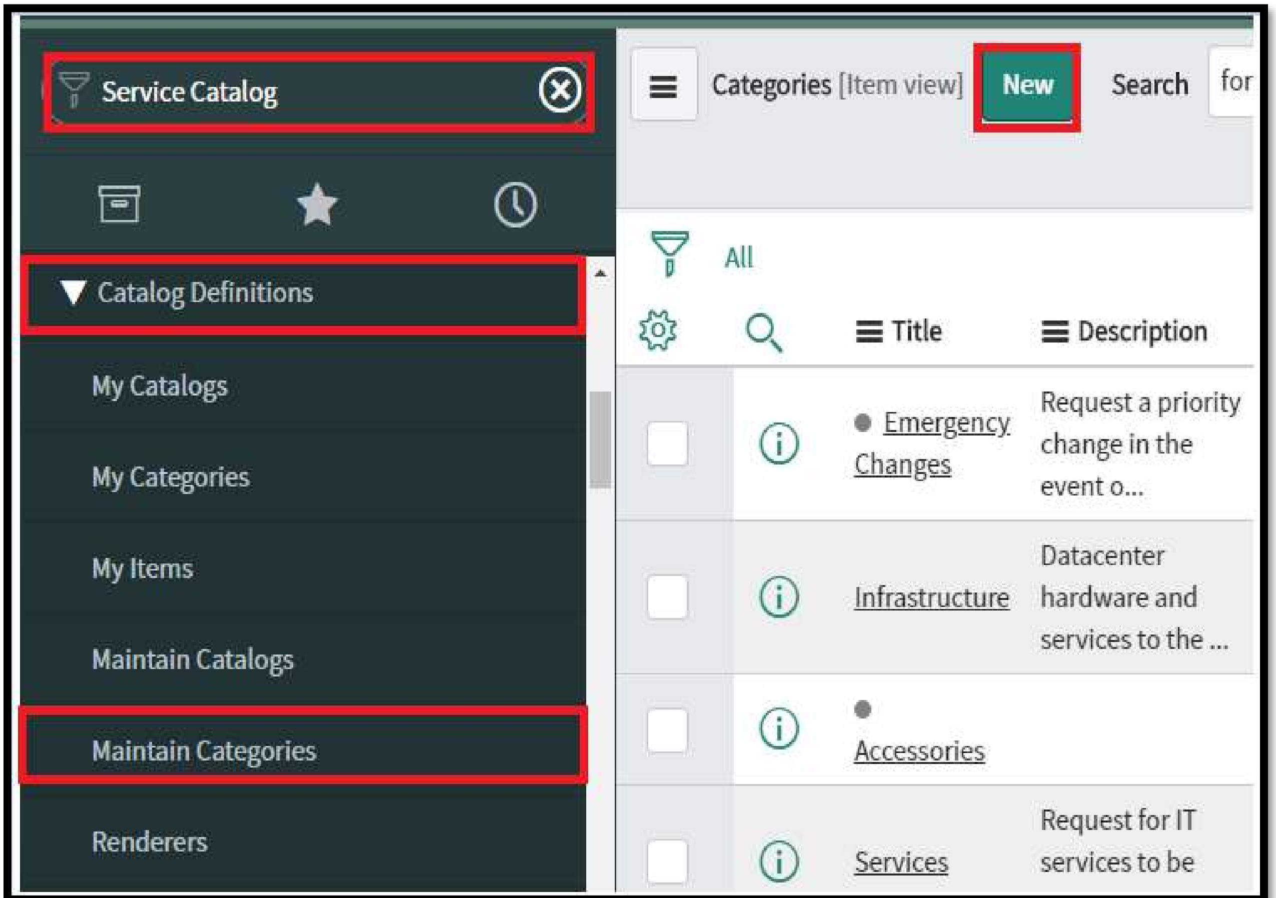
Exercise: 3 Configure Service Catalog Category (Wood Table)

Configuring new **Service Catalog Category**

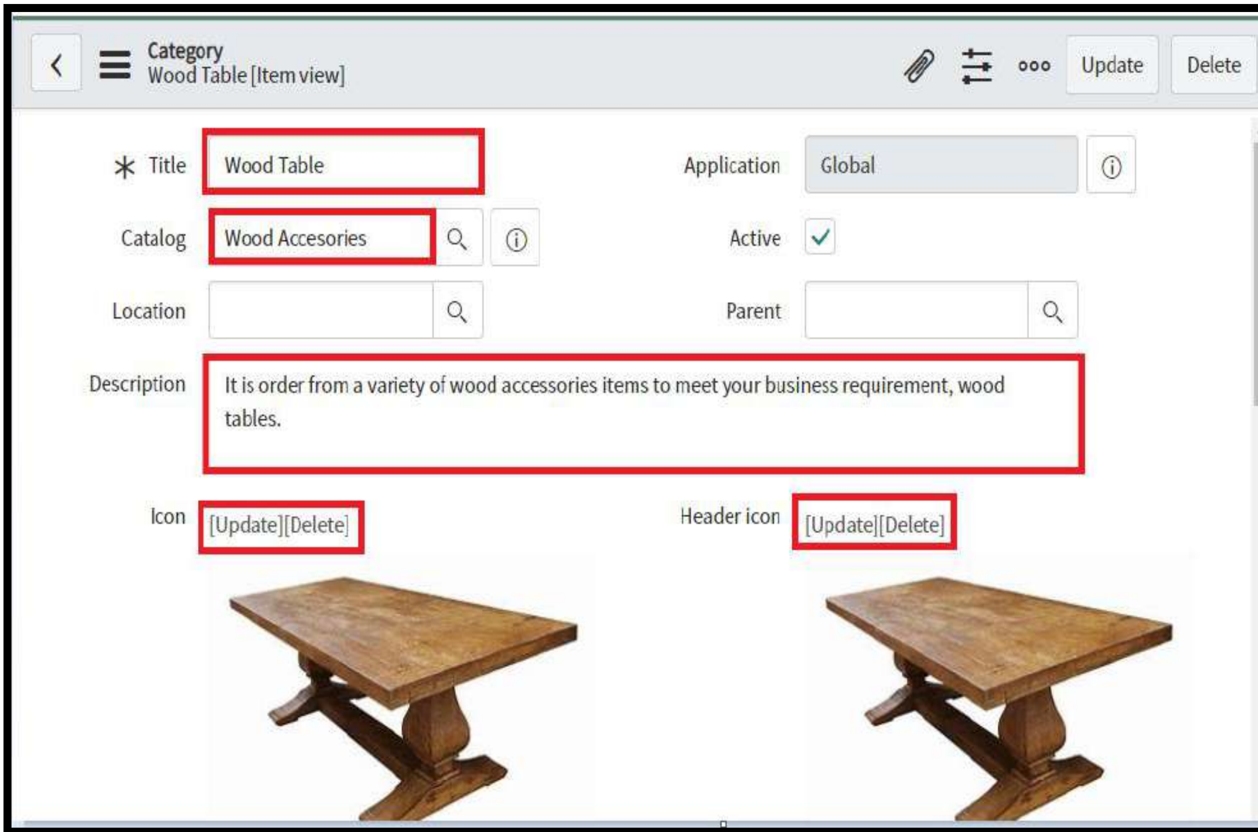
Category: Wood Table

Procedure

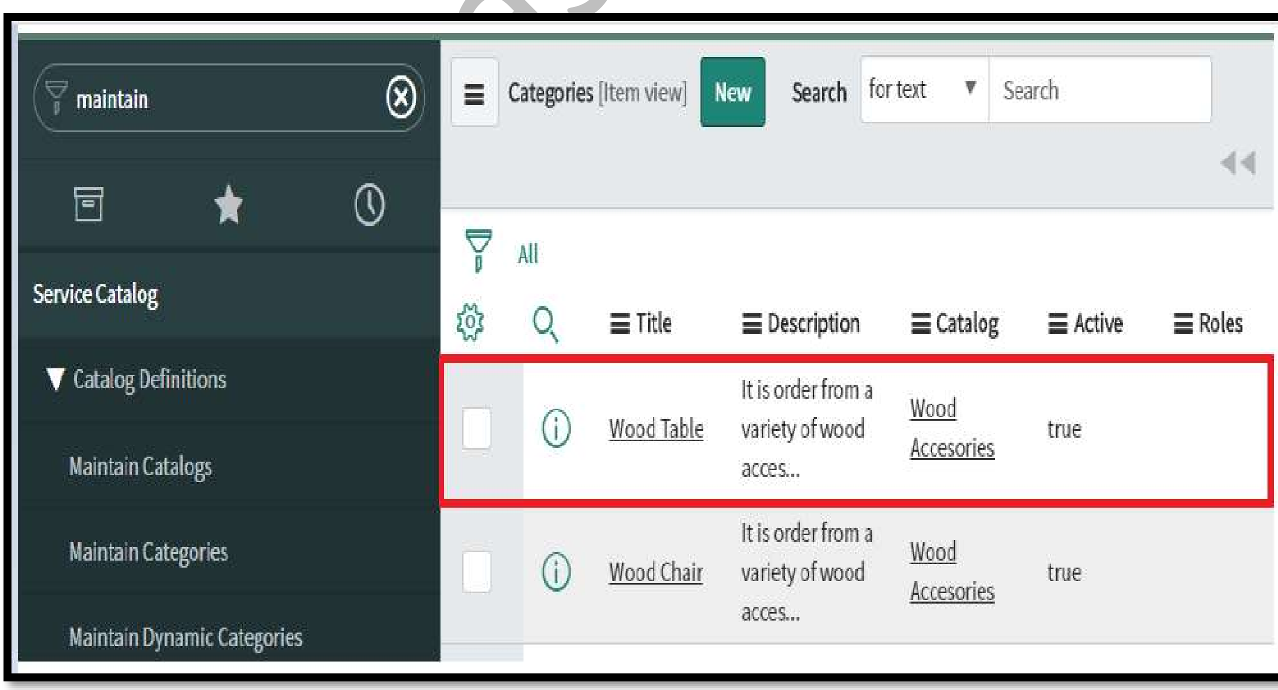
1. Navigate to **Service Catalog > Catalog Definition > Maintain Categories**
2. Click on **New**
3. Fill the **Service Catalog Categories** Form



4. **Title:** Wood Table
5. **Catalog:** Wood Accessories
6. **Location:** Optional
7. **Description:** It is order from a variety of wood accessories items to meet your business requirement, **wood tables.**
8. **Icon:** Add
9. **Header Icon:** Add
10. Click on **Submit**

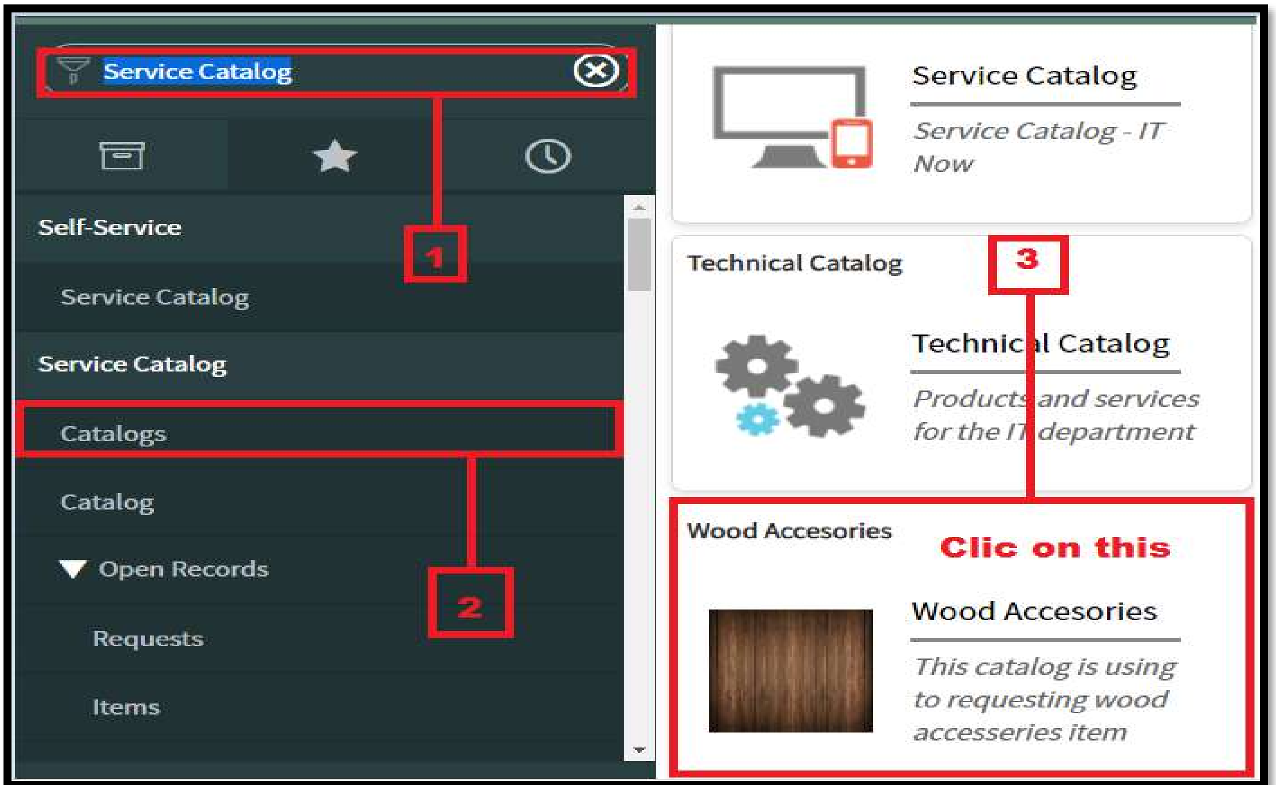


11. Successfully configured new **Category** into list of list of **Categories**

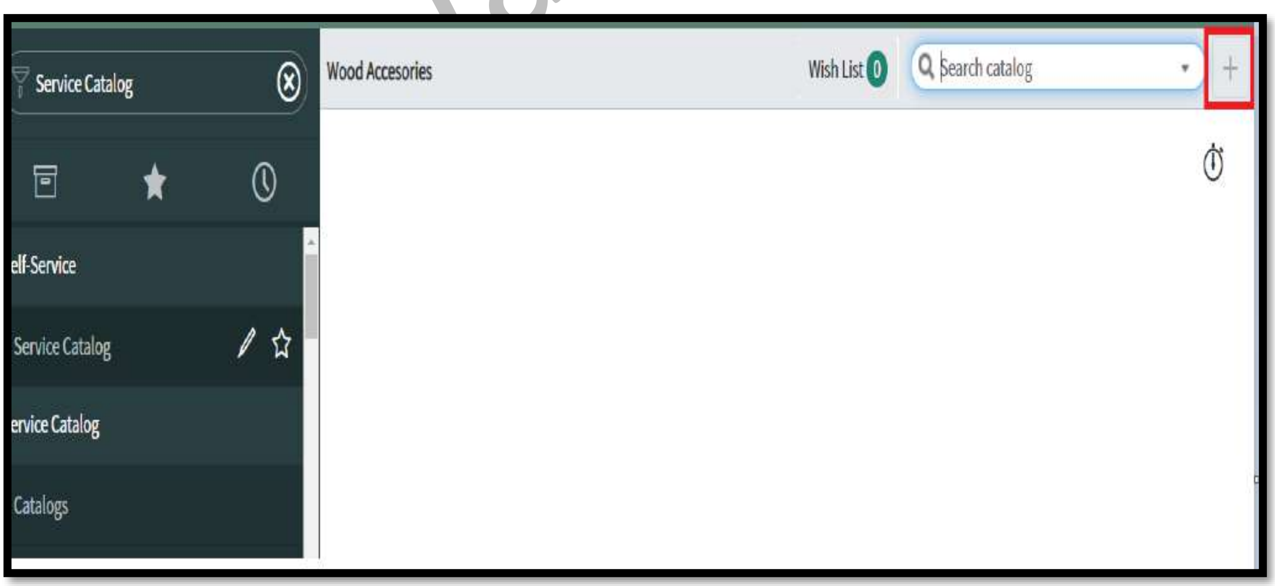


12. Add this new **Category** to category **Home Page**

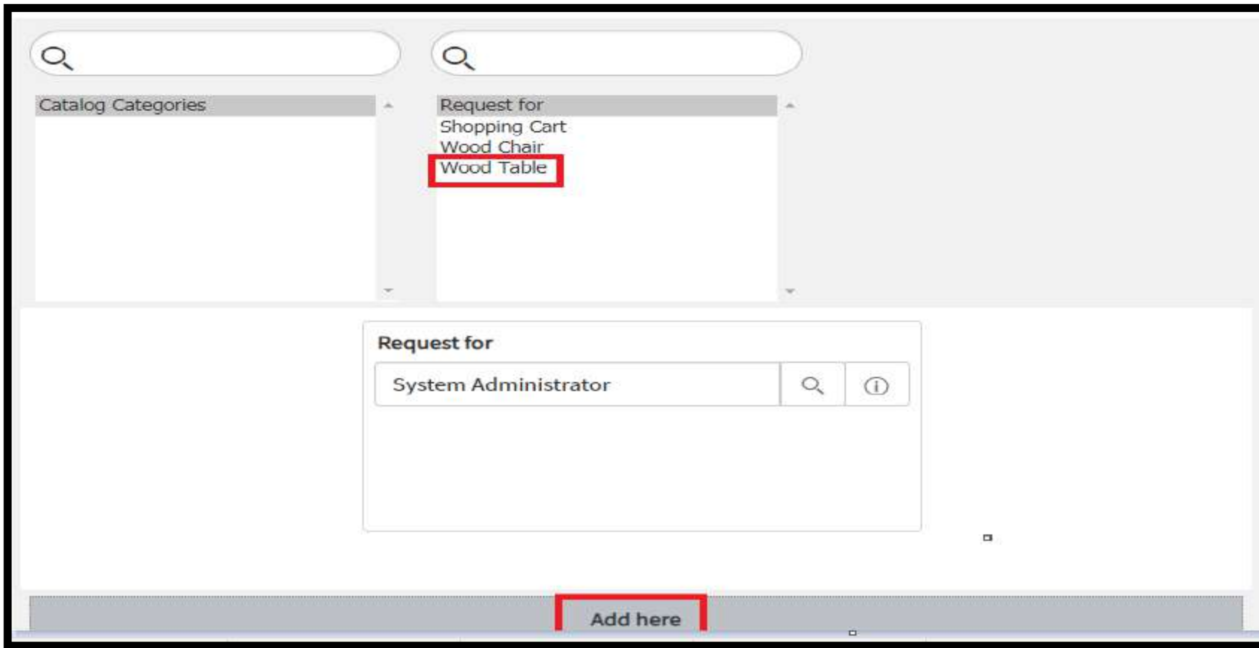
13. Navigate to **Service Catalog > Catalogs > Wood Accessories**



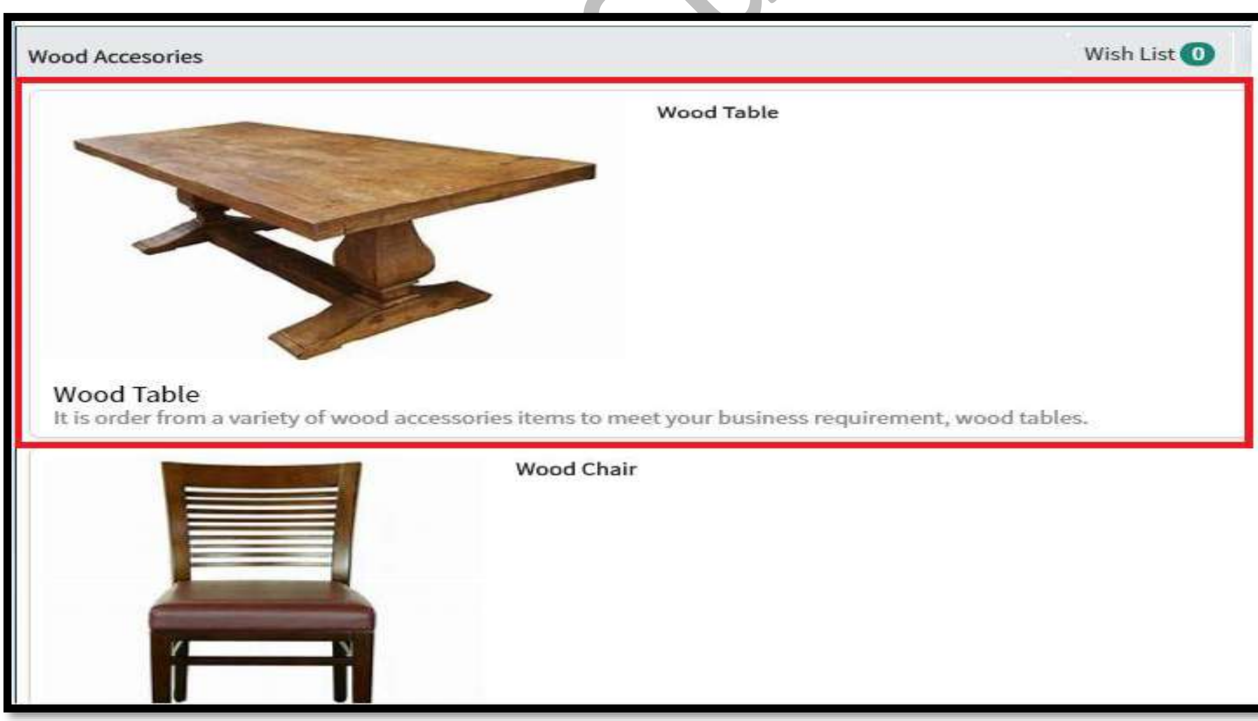
14. Click on **+ icon**



15. Select **Wood Chair**
16. Click on **Add here**



17. Successfully added new **Service Category** (Like - Below screenshot)

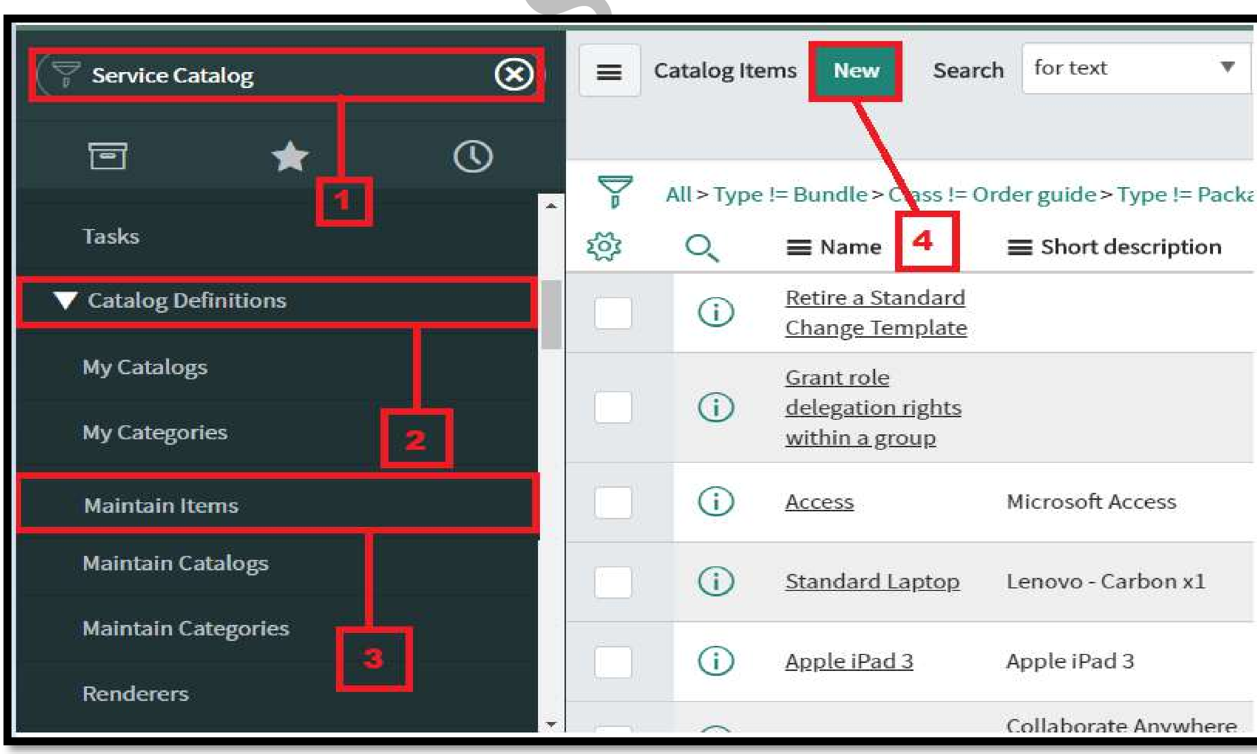


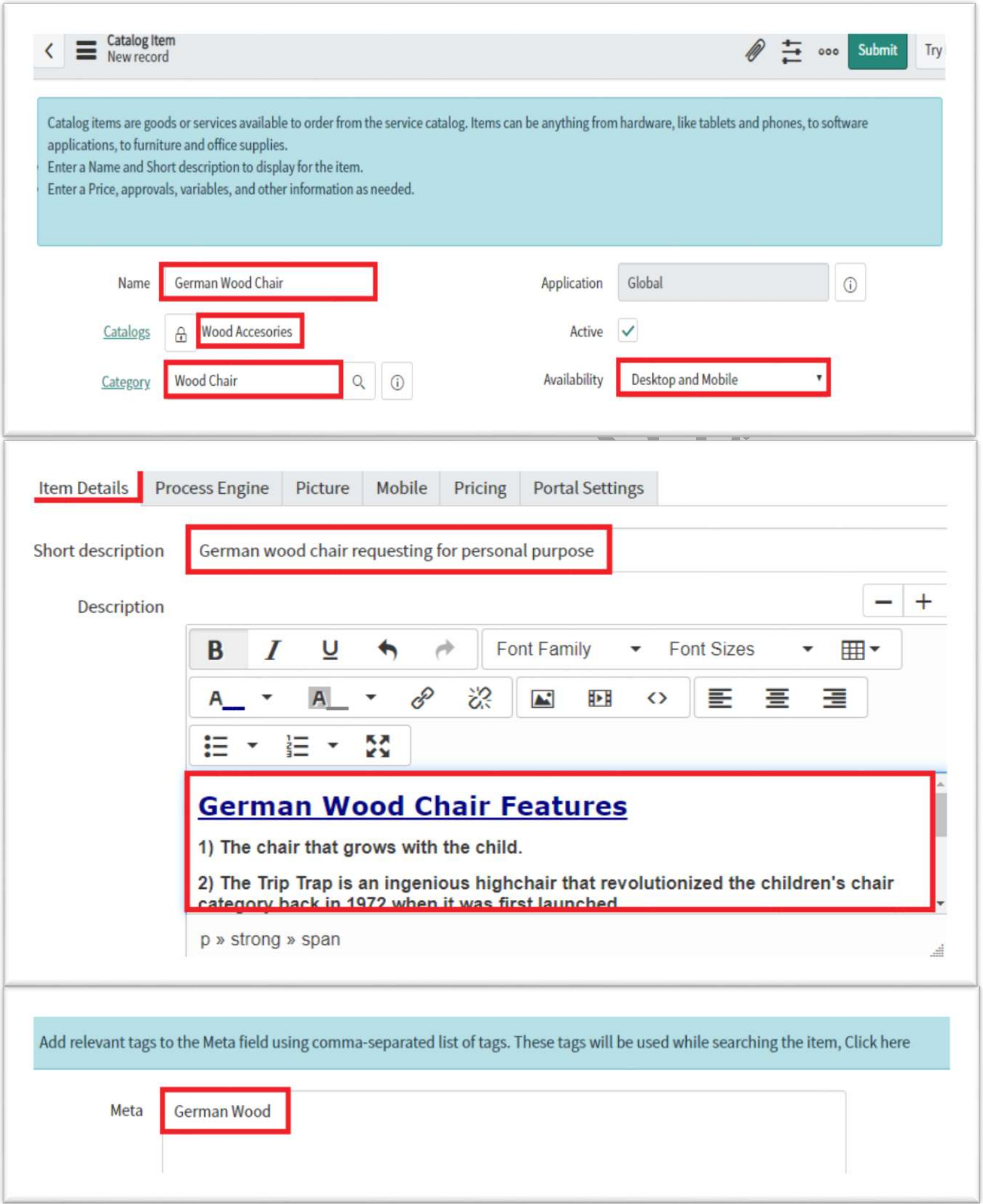
Exercise: 4 Configure Service Catalog Item

1. **Service Catalog Item** can be defining an **Item or service**. If any customer something can be ordered by itself, it is a catalog item.
2. **Ex: Apple Macpro 15** is a catalog item, that can be requested by delegates
3. It can be used to publish a service to all our internal users in organization. Then we can add more details about item like. an **Item description, images, workflow engine**, **Price details, recurring price** to determine the approval and fulfillment processes for the catalog items.
4. We can use different type Variables in service catalog item to present and get more information from the users.
5. Catalog **UI Policies** and **Catalog Client Scripts** can also be added to control the item behavior based on user input.
6. And we can configure **User Criteria** to display item to respective users

Procedure

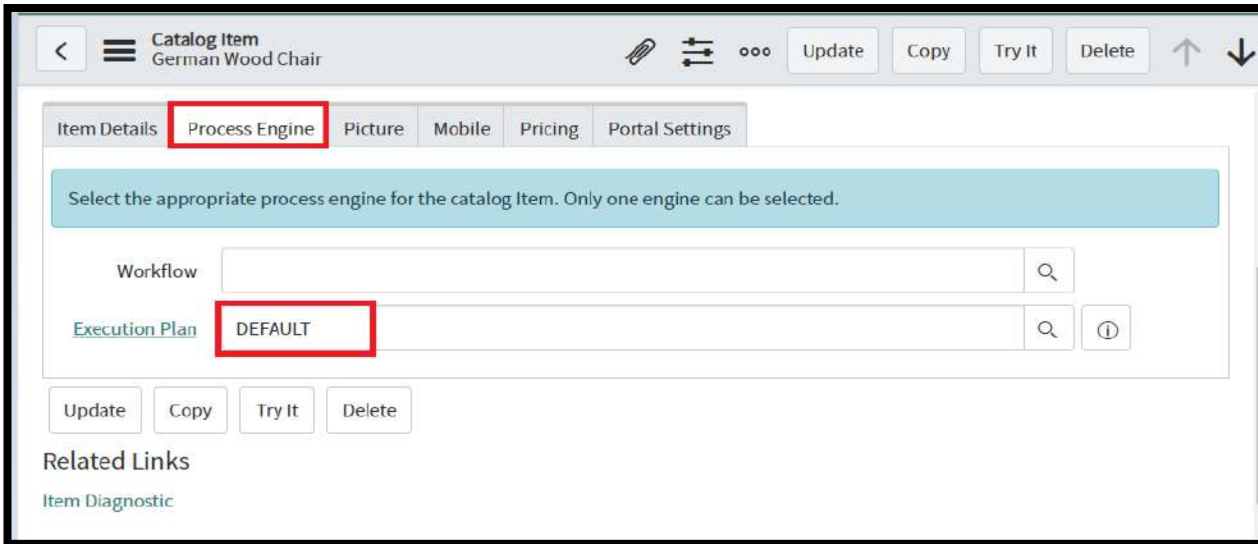
1. Navigate to **Service Catalog > Catalog Definition > Maintain Item**
2. Click on **New**
3. Fill the **Service Catalog Item** Form



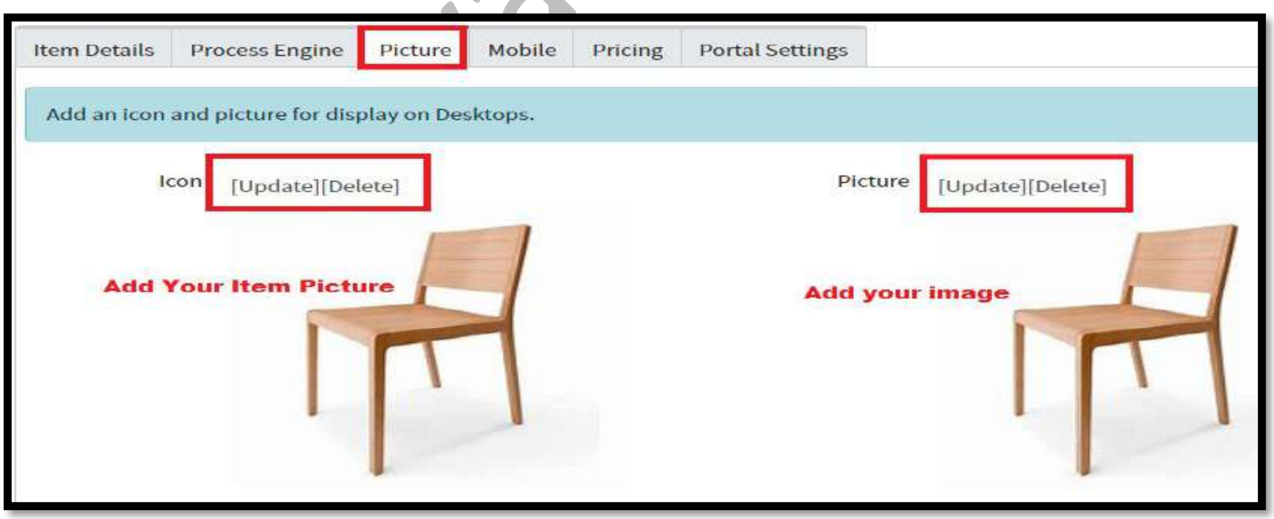


4. **Name:** German Wood Chair

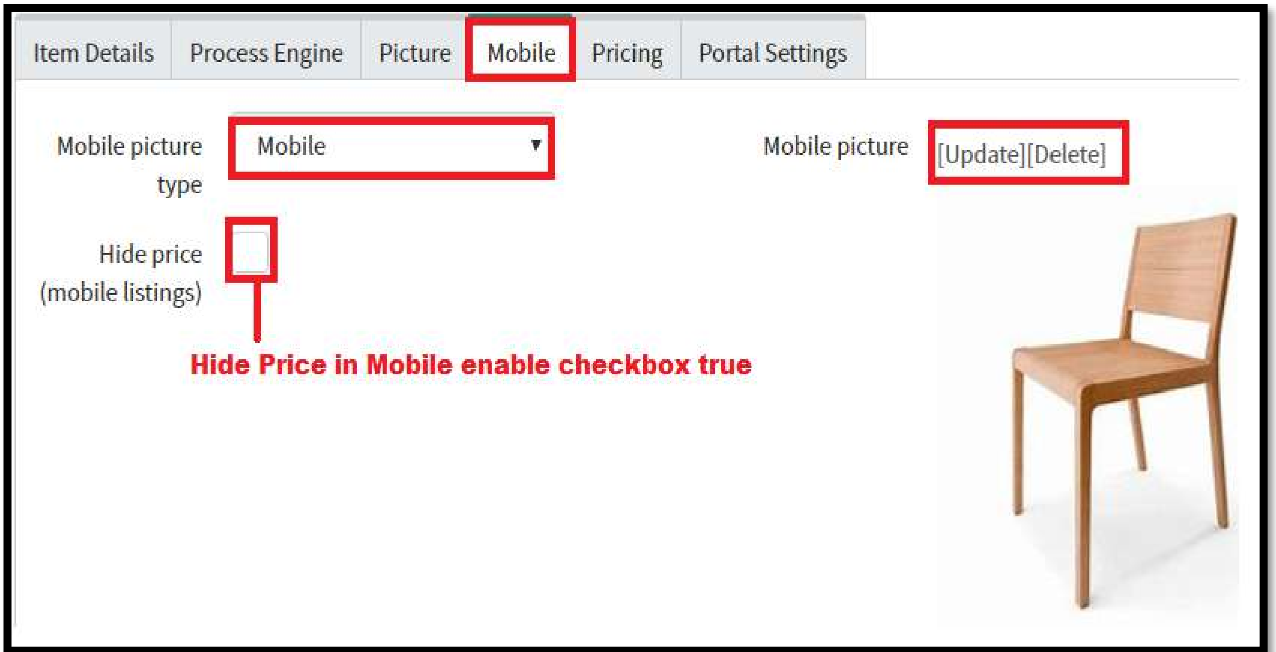
5. **Catalog:** Wood Accessories
6. **Category:** Wood Chair
7. **Availability:** Desktop and Mobile
8. **Short Description:** German wood chair requesting for personal purpose
9. **Meta:** German Wood



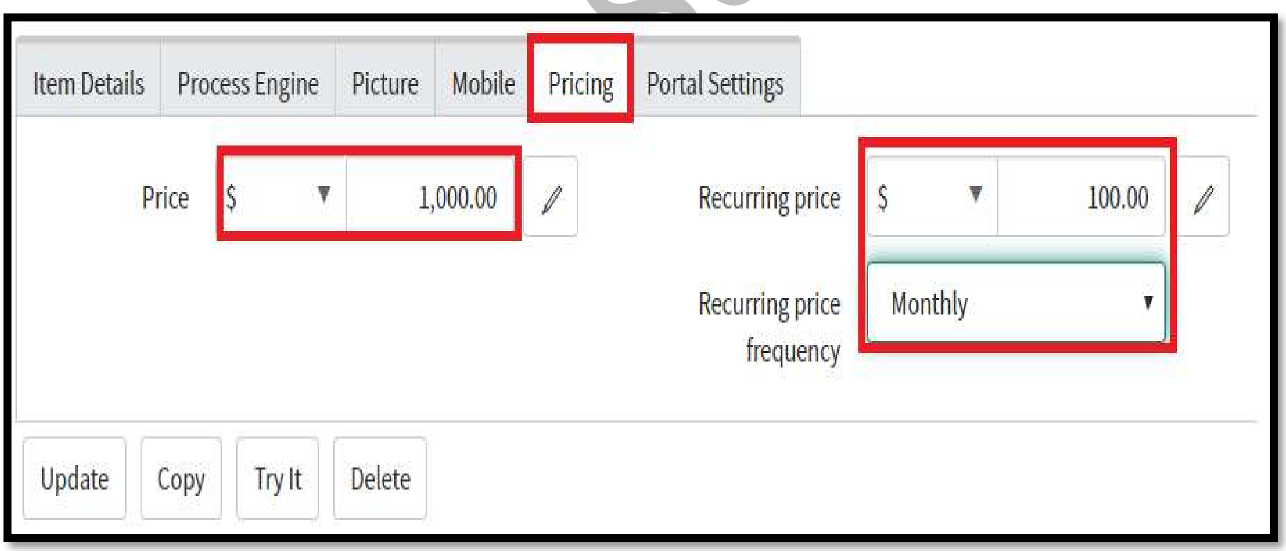
10. **Open:** Process Engine Tab
11. **Workflow:** Left blank
12. **Execution Plan:** DEFAULT



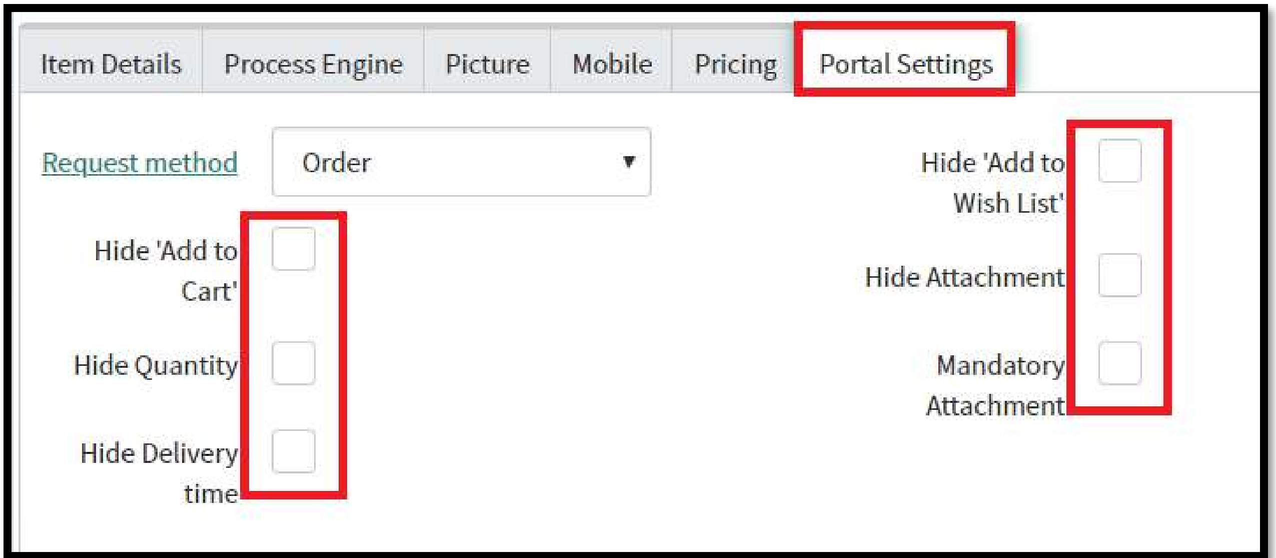
13. **Open:** Picture Tab
14. **Icon:** Add
15. **Picture:** Add



16. **Open:** Mobile Tab
17. **Mobile Picture Type:** Mobile



18. **Open:** Pricing Tab
19. **Price:** 1,000 $
20. **Recurring Price:** 100$
21. **Recurring Price Frequency:** Monthly



22. Open: Portal Settings

Note: Based on Our client requirement we need to enable these all checkboxes

Types of Catalog Items

Service Catalog Application offering different types of catalog items

Record producers:	✓ Creating **Task Based** record instead of creating incident
Order guide	✓ To requesting bulk of items in **Single Request**
Content Items:	✓ **Service Catalog Items** provide the information instead of **Goods or Services.**

What Record Producer

A record producer is a kind of catalog item that can allows end users to create task-based records, such as incident records, from the service catalog. It is alternate way create incident record

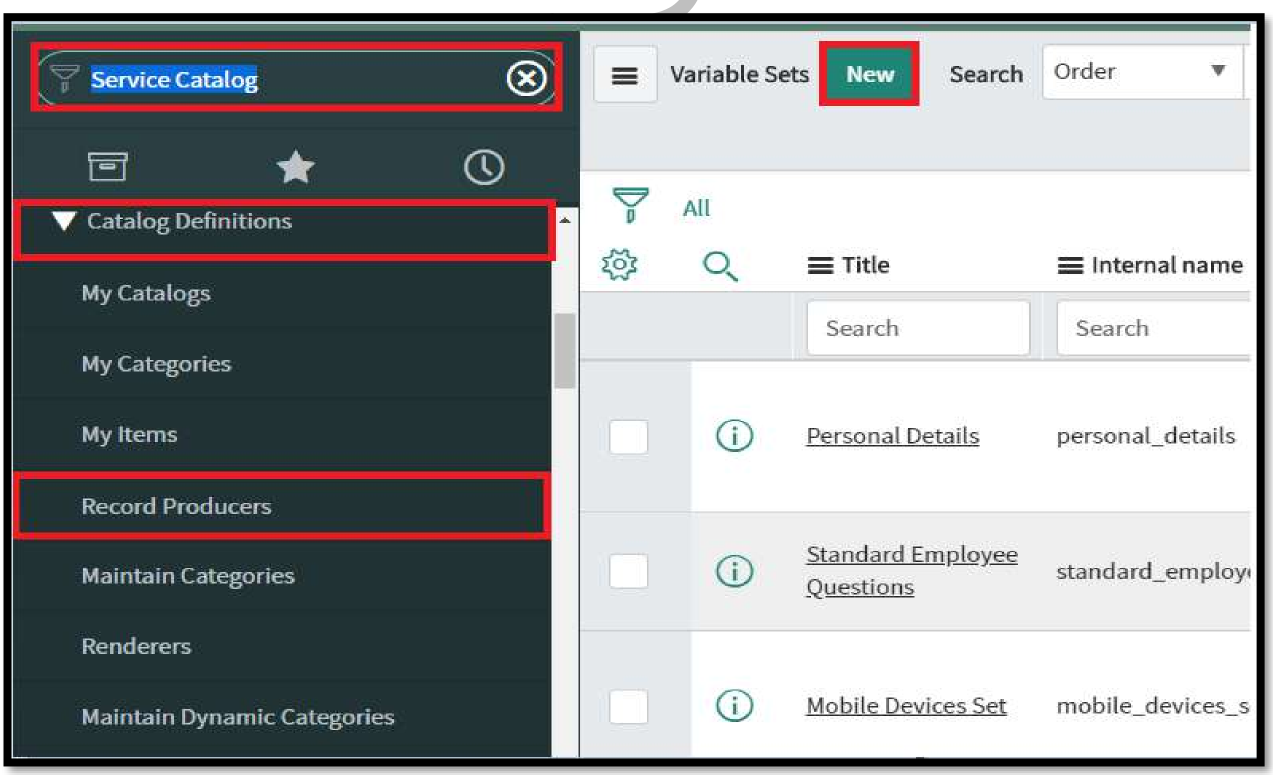
Record Producer can provide better **Customer** experience instead of following the regular task-based form for creating records. The look and feel of a record producer is similar to that of a catalog item. But the record producer **generates a task record such as incident, instead of a requested item.**

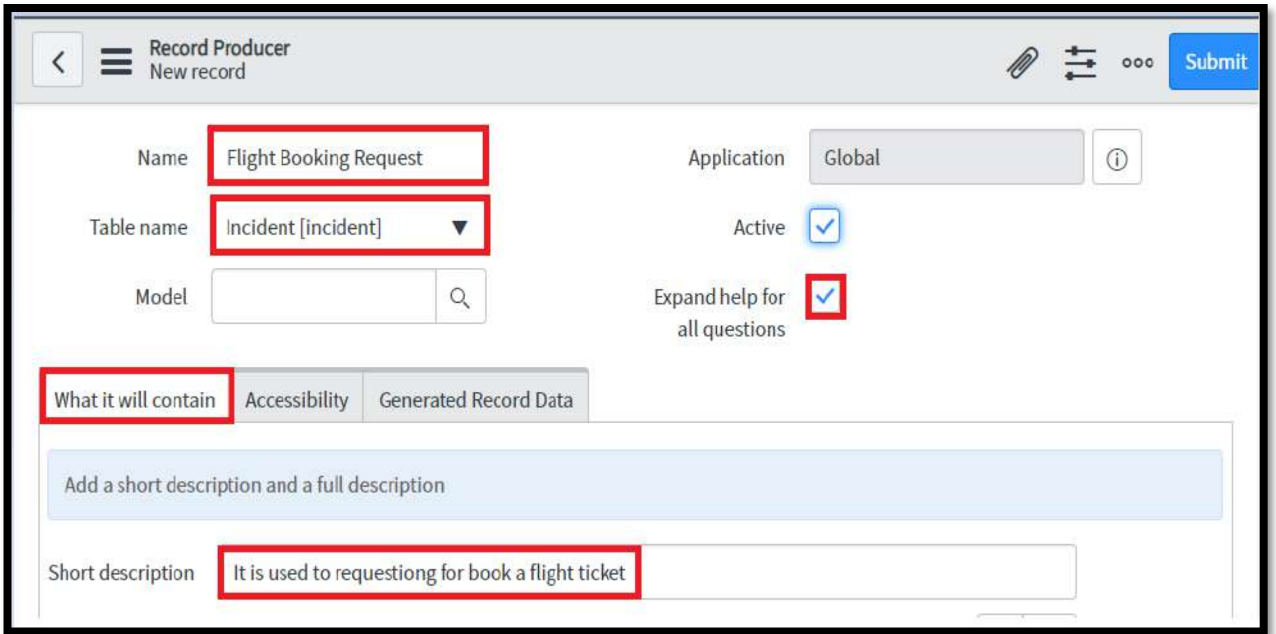
Exercise :5 Crete Record Producer

We can create a record producer for tables and database views that are in the same scope as the record producer. Also for tables that allow create access from applications in other scopes.

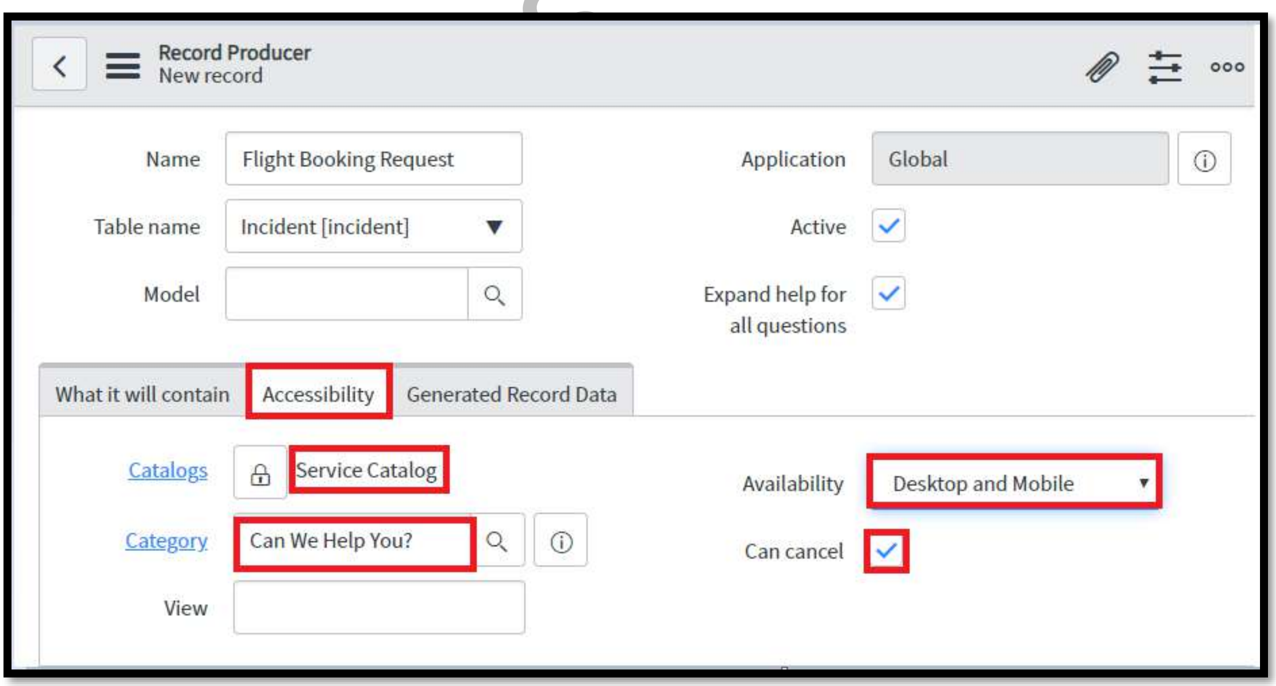
Procedure

1. Navigate to **Service Catalog > Catalog Definition > Record Producers.**
2. Click on **New**
3. Fill the **Record Producer** form.



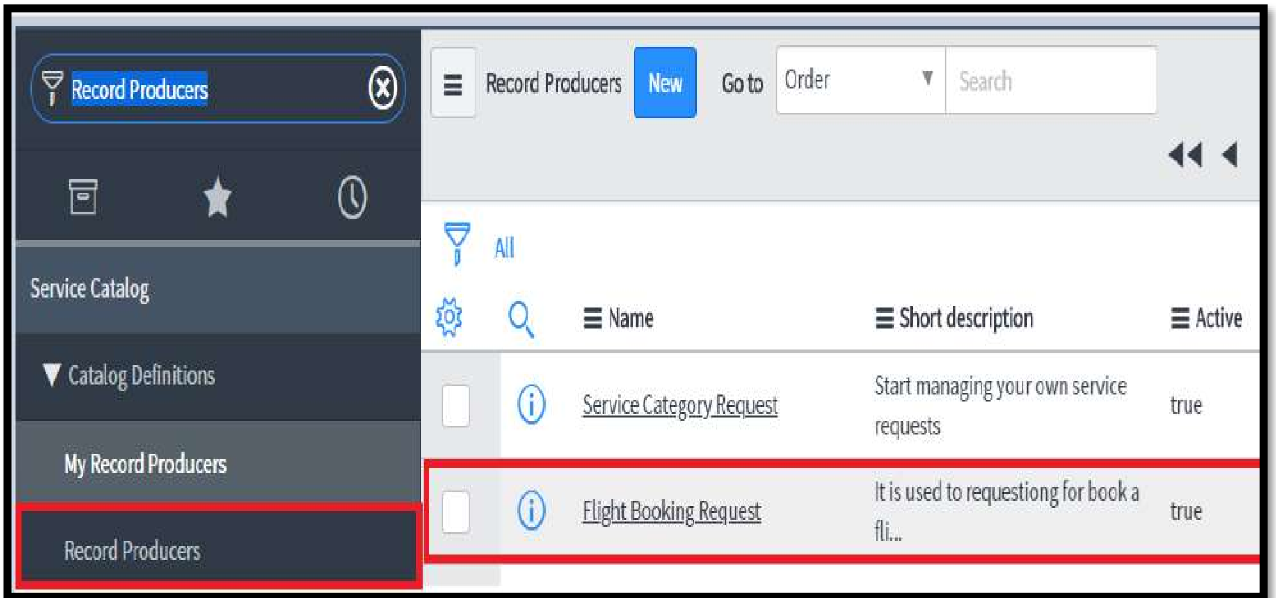


4. **Name:** Flight Booking Request
5. **Table Name:** Incident
6. **Expand help for all questions:** True
7. **Short Description:** It is used to requesting for book a flight ticket



8. Open **Accessibility** Section
9. **Catalogs:** Service Catalog
10. **Category:** Can we help you?

11. Availability: Desktop and Mobile
12. Can Cancel: Check **true**
13. Click on **Submit**



14. Consider above screenshot after created **Record Producer**

Working with Variable

What is Variable

Service Catalog Variables can be collecting and passing the information about the customer makes when ordering a particular catalog item or raise a request through record producer. Variables can help define the structure of a catalog item form that is displayed to the customer about the order and status.

For example, any customer requested an **I Phone** mobile, customer can be able to select more options like **Mobile Color, Extra memory, accessories**

Note: Based on your business requirement the service now developer or admin can be able to configure variables

Types of service catalog variables

Service Catalog offering several types of variables, which are also referred to as questions. Variables can be stored, accessed from multiple places, and passed between tasks in a process when fulfilling a request. They can be displayed on the Requested Item and Catalog Task forms after an item has been ordered.

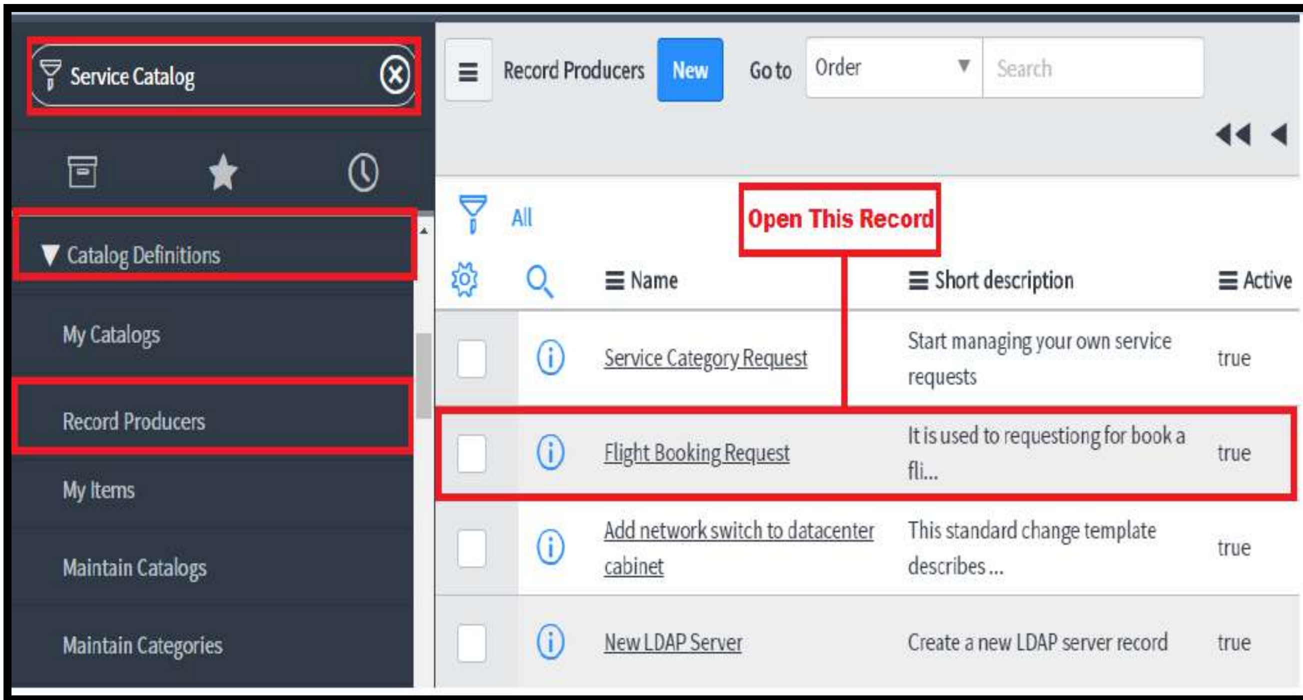
Service Catalog Variables	
✓ Break	✓ List Collector
✓ Container Start	✓ Check Box
✓ Container Split	✓ Reference
✓ Container End	✓ Select Box
✓ Duration	✓ Lookup Multiple Choice
✓ Yes/No	✓ Macro
✓ Numeric Scale	✓ Wide Single line text
✓ Email	✓ Macro with Label
✓ HTML	✓ Masked
✓ Label	✓ Multiple Choice
✓ Multi Line Text	✓ UI Page
✓ Single Line Text	✓ Lookup Select Box
✓ URL	✓ Date
	✓ Date and Time

Exercise: 6 Create Variables in Record Producers

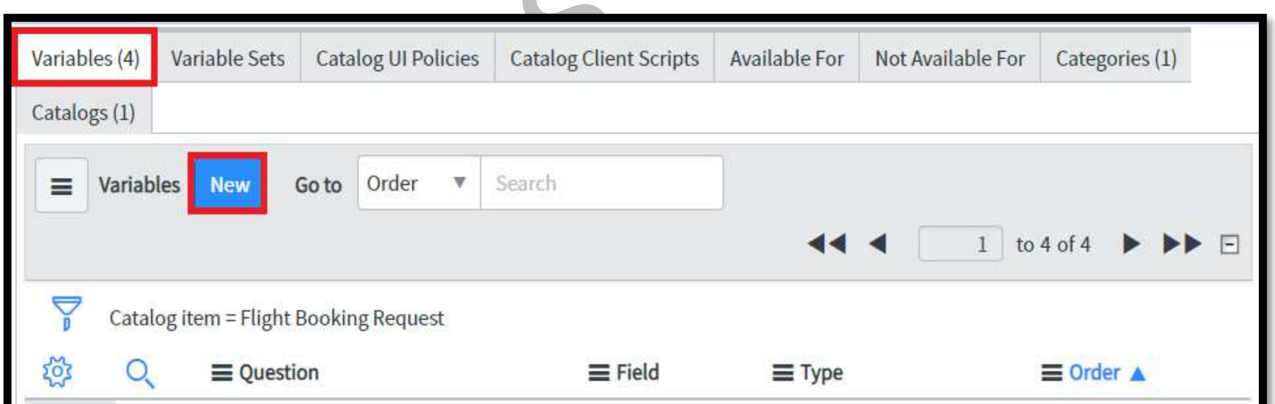
Create variables in our existing record producer **(Flight Booking Request)**

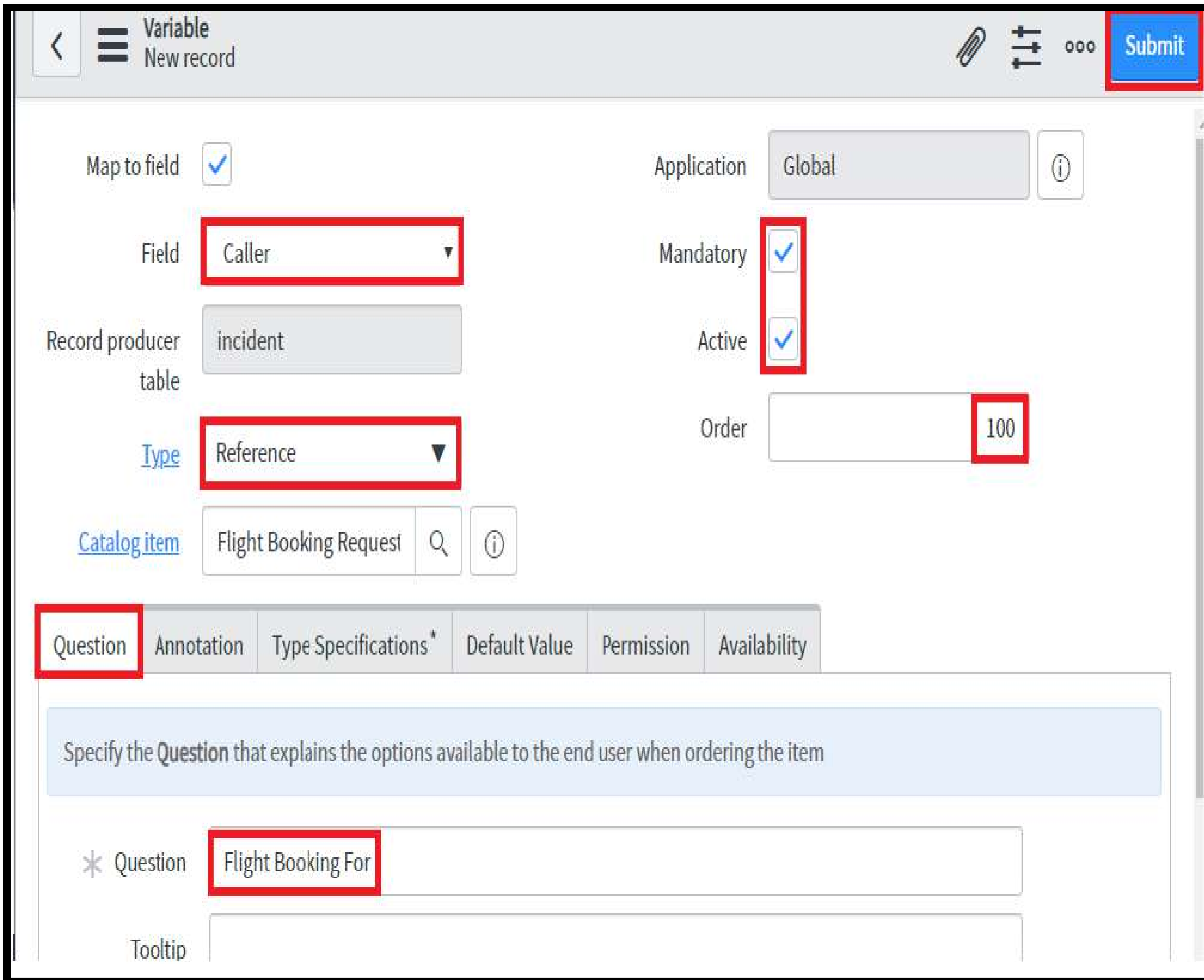
Procedure

1. Navigate to **Service Catalog > Catalog Definition > Record Producers**
2. Open **Flight Booking Request** Record Producer



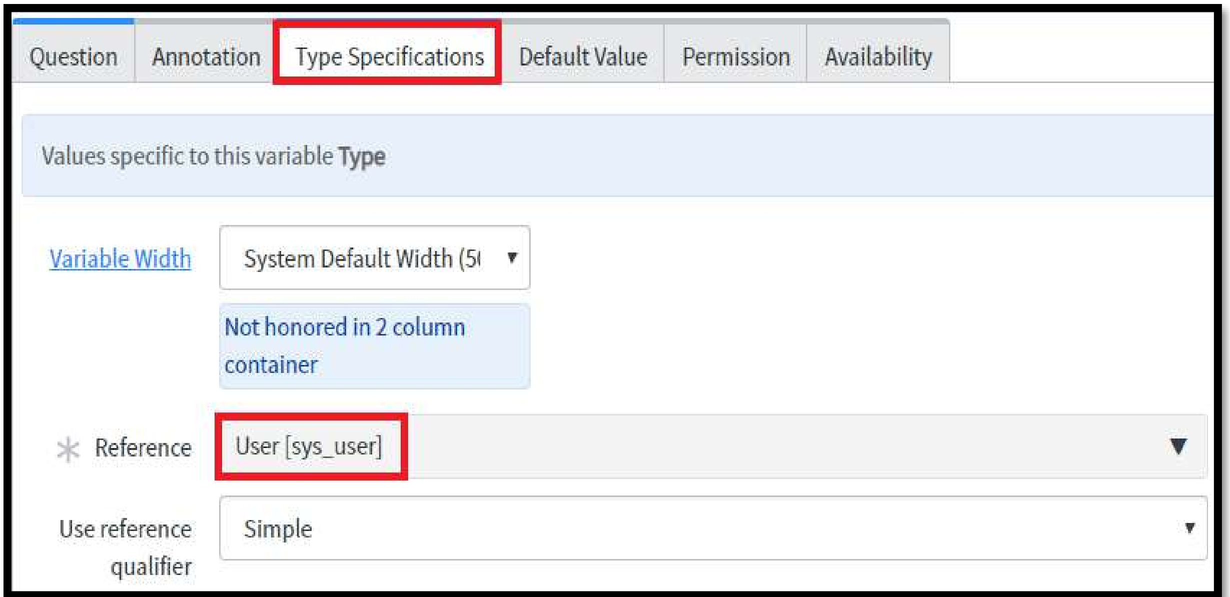
3. Open **Variable Tab** from Related List
4. Click on **New**
5. Fill the **Reference Variable** form



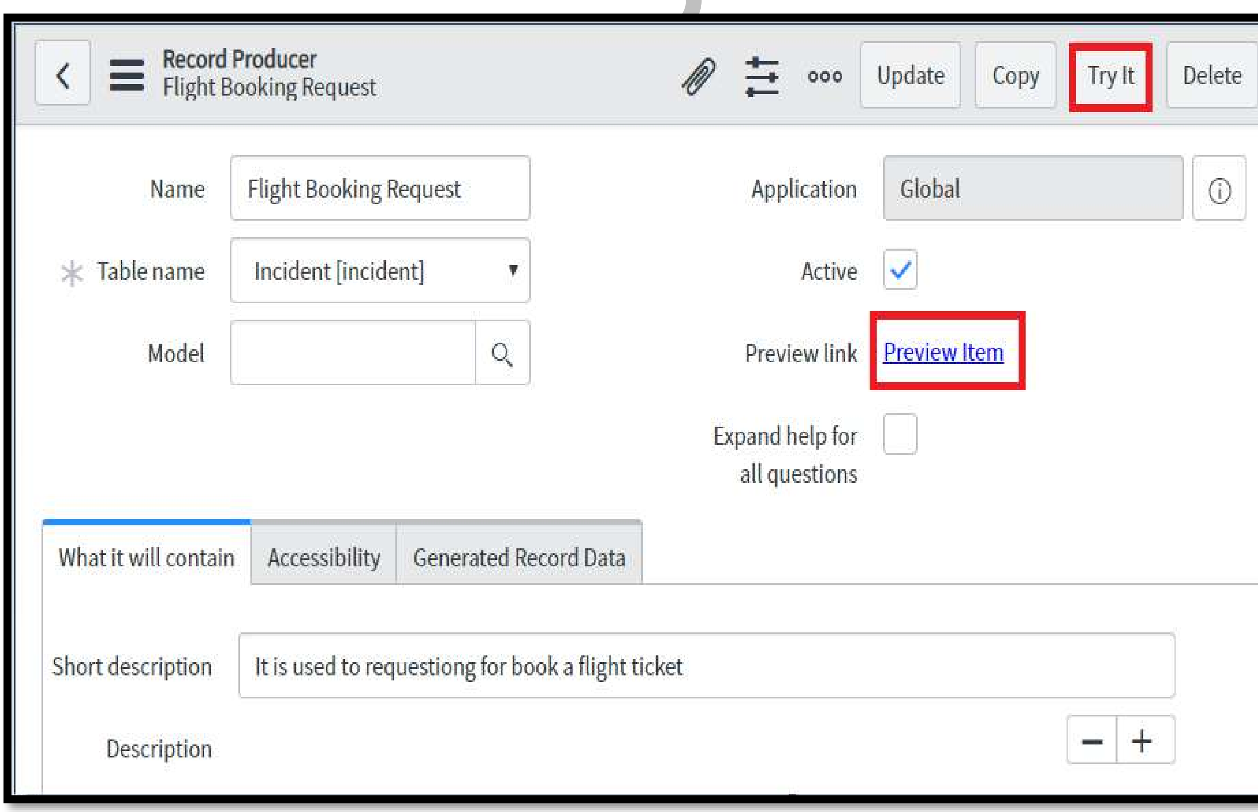


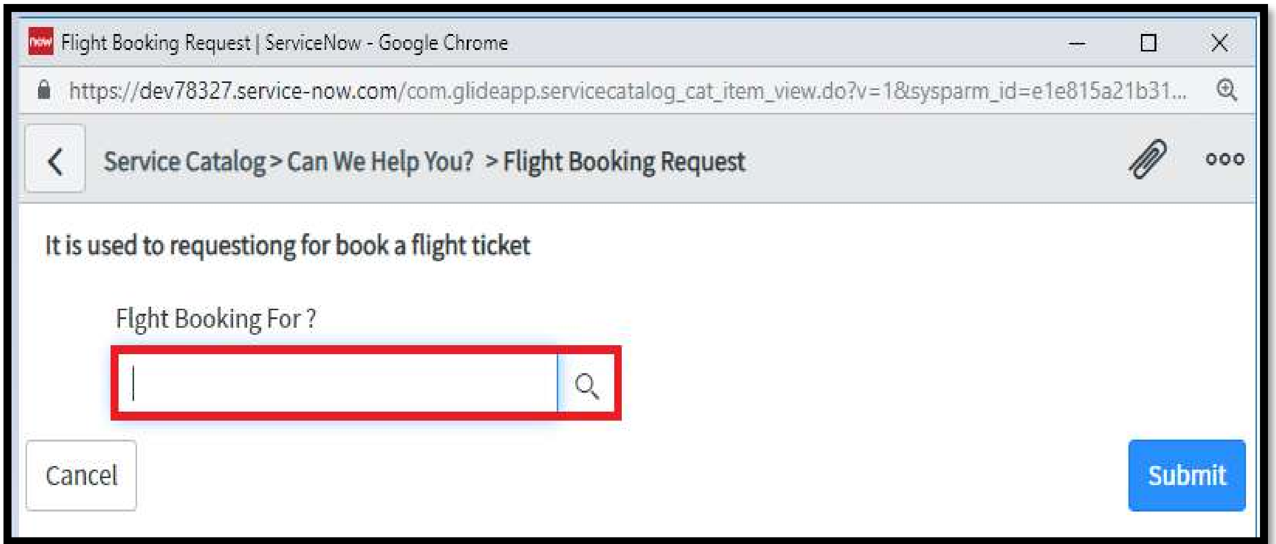
6. **Map to field:** Check True **(This variable value can store in caller field in incident table)**
7. **Record Producer Table:** Incident **(Records in store in Incident table)**
8. **Field:** Caller
9. **Mandatory:** Check True **(customer must fill the value)**
10. **Variable Type:** Reference **(Data type)**
11. **Order:** 100
12. **Question:** Flight Boking For

Note: Reference Variable is used to get the records from reference table like **sys_user**



13. Open **Type Specifications** tab from related list
14. Reference: [sys_user]
15. Click on **Submit**
16. Click on **Preview Item or Try It** to preview the record producer





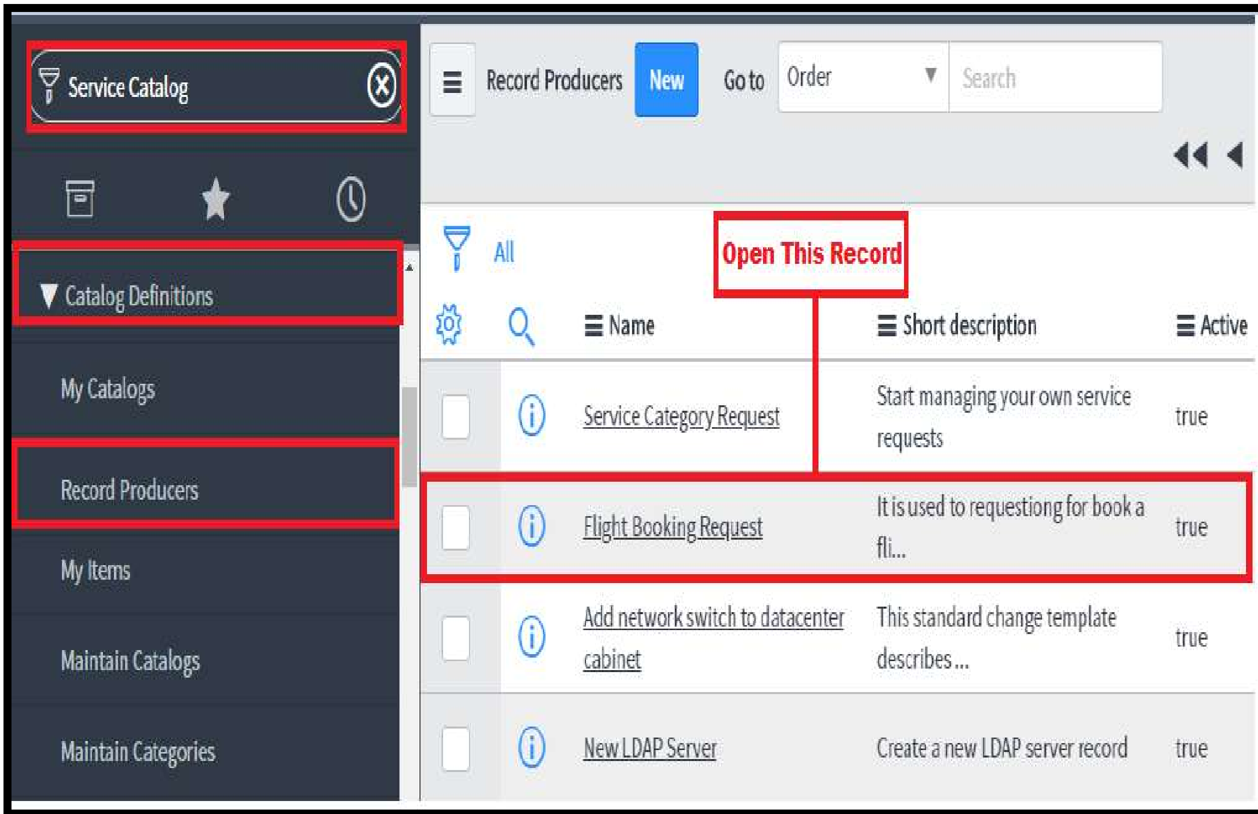
17. Preview of **Record Producer Variable**

Working with Yes/No variable

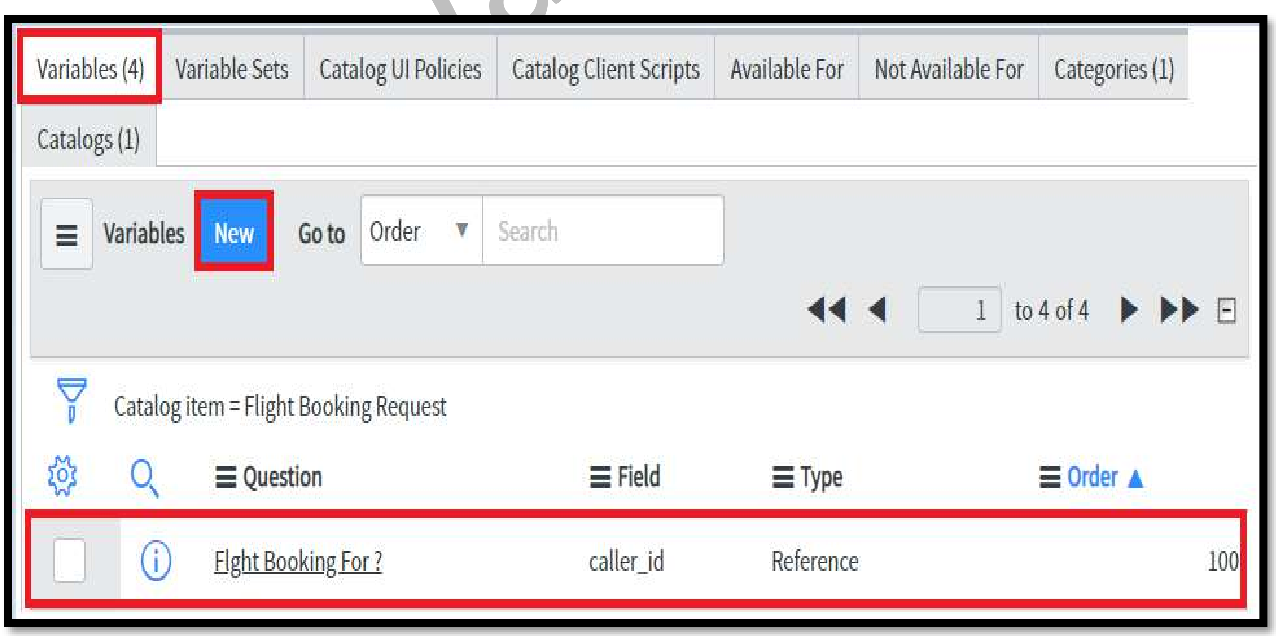
The Yes or No variable creates a choice list or boo lions with **Yes / No** as options. Include None also if required

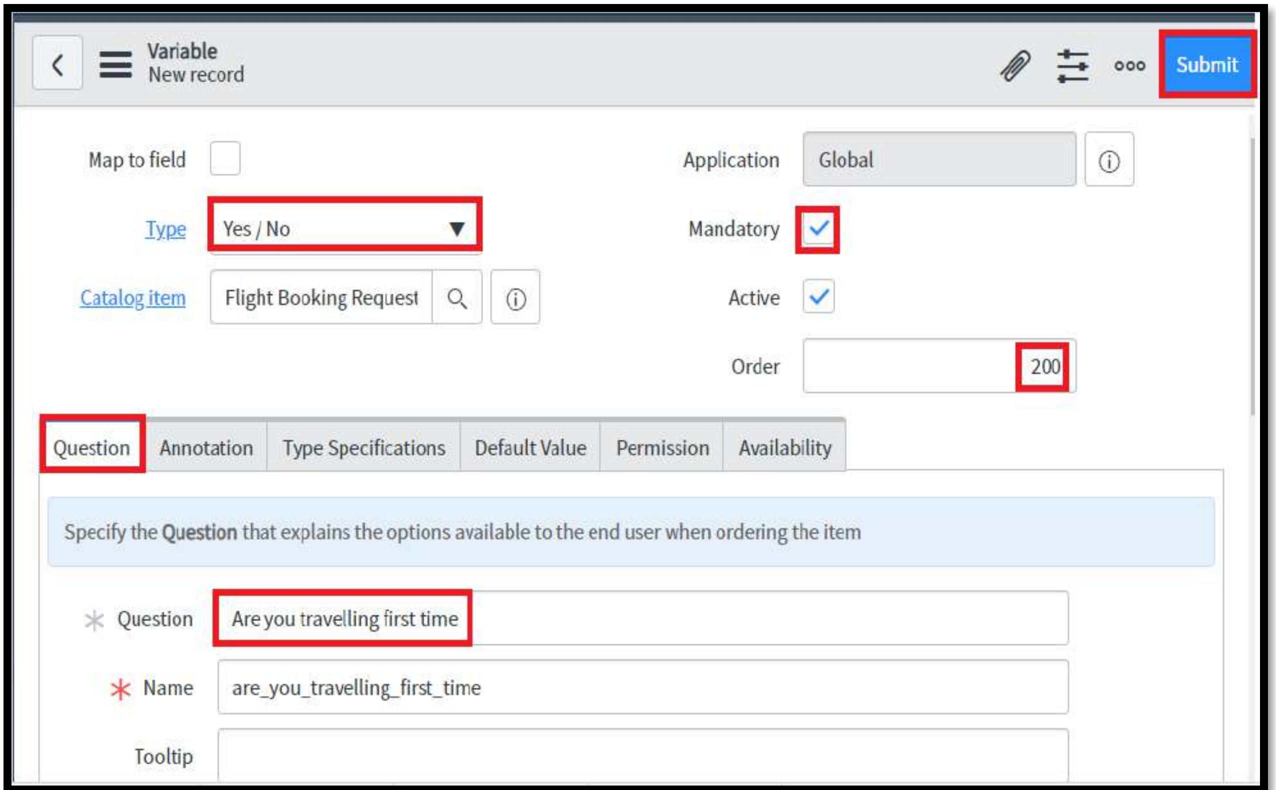
Procedure

1. Open **Flight Booking Request** Record Producer

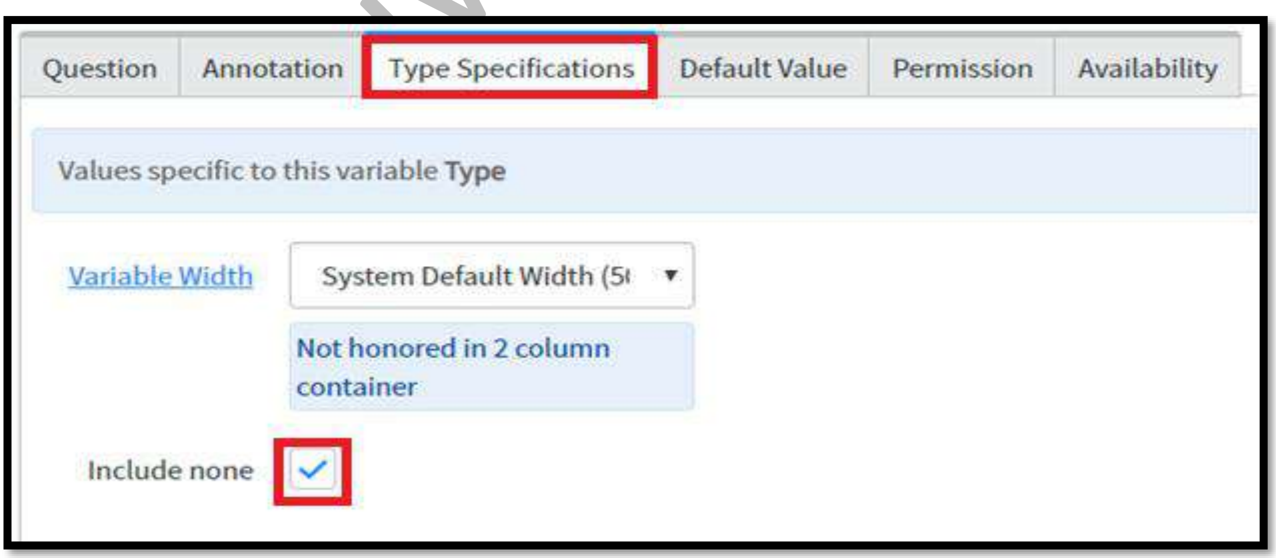


2. Open **Variable Tab** from Related List
3. Click on **New**
4. Fill the **Yes/No Variable** form



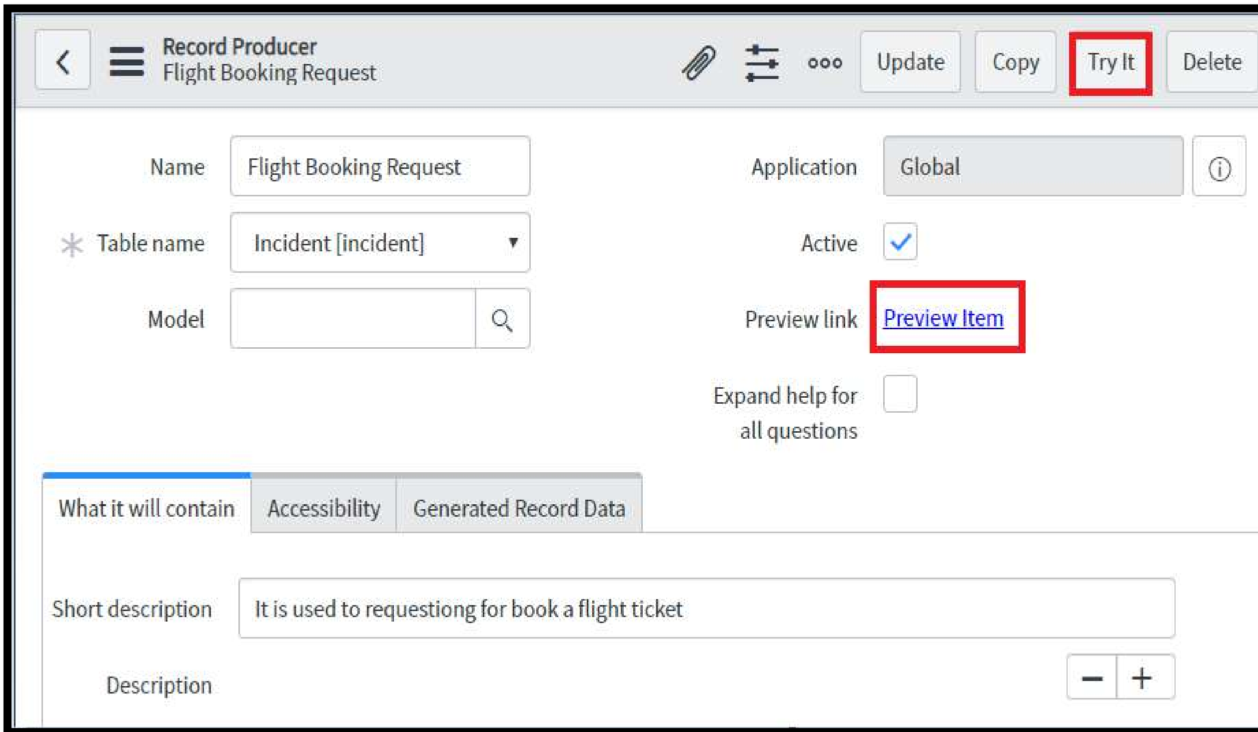


5. **Type:** Yes/No
6. **Mandatory: Check True**
7. **Order:** 200
8. **Question:** Are you travelling first time

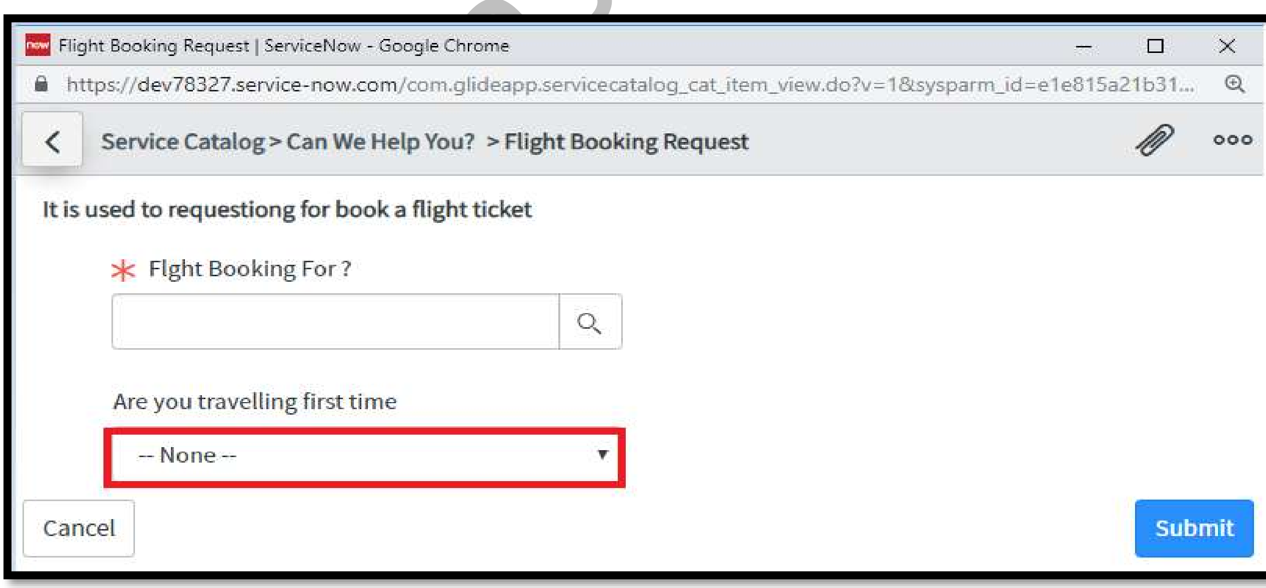


9. **Type Specification Tab**
10. **Include none**: checked **True**

11. Click on **Submit**



12. Click on **Preview Item or Try It** to preview the record producer



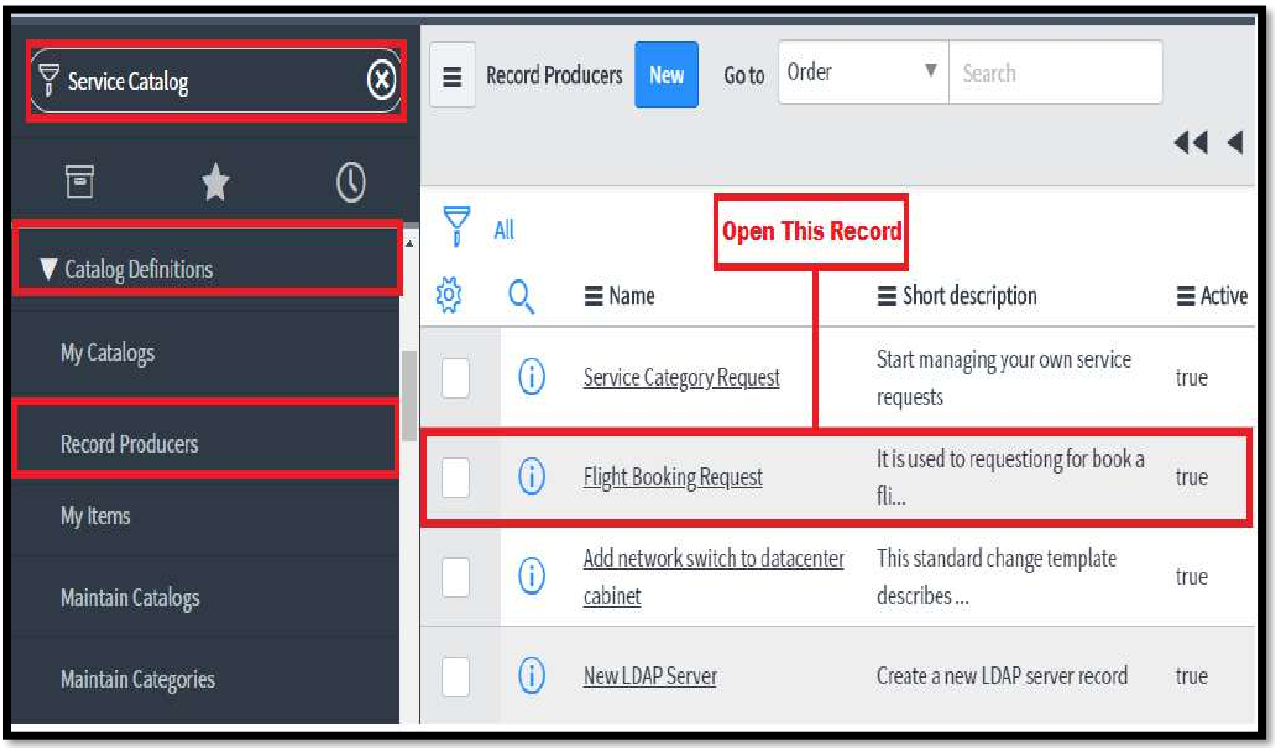
13. Preview of **Record Producer Variable**

Working with Lookup Select Box Variable

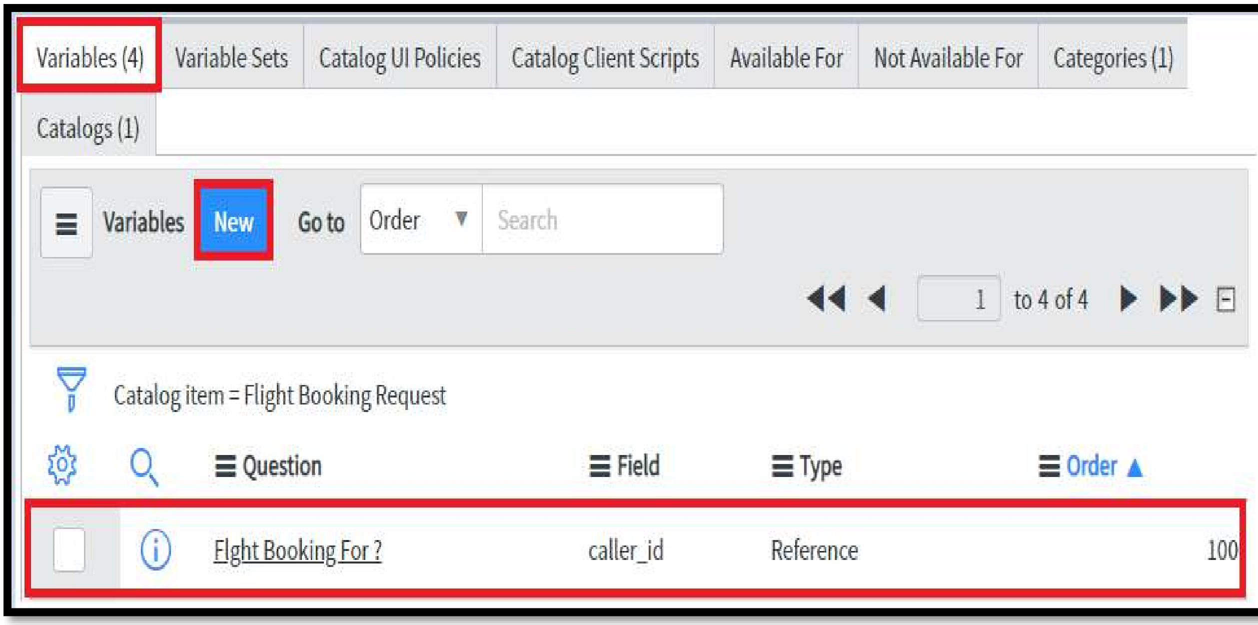
We can get choice list using data queried from a table. Functionality is similar to **lookup multiple choice variable**, which creates radio buttons from queried data.

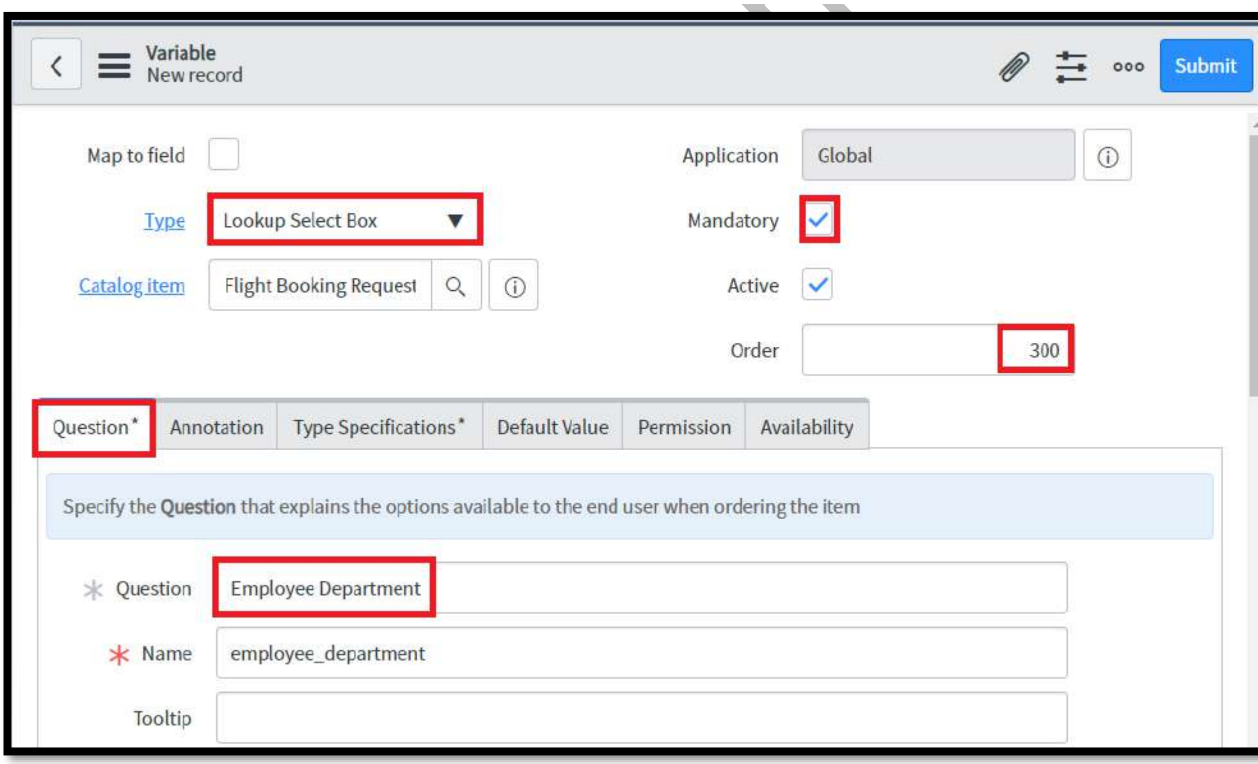
Procedure

1. Open **Flight Booking Request** Record Producer

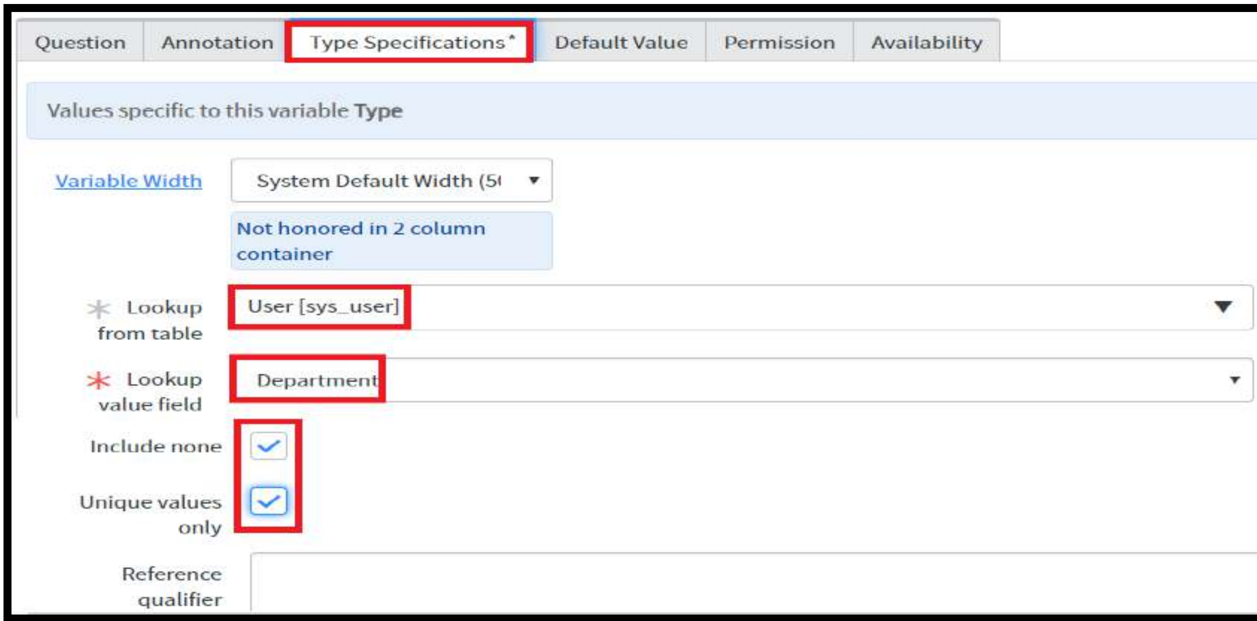


2. Open **Variable Tab** from Related List
3. Click on **New**
4. Fill the **Lookup Select Box** Variable form

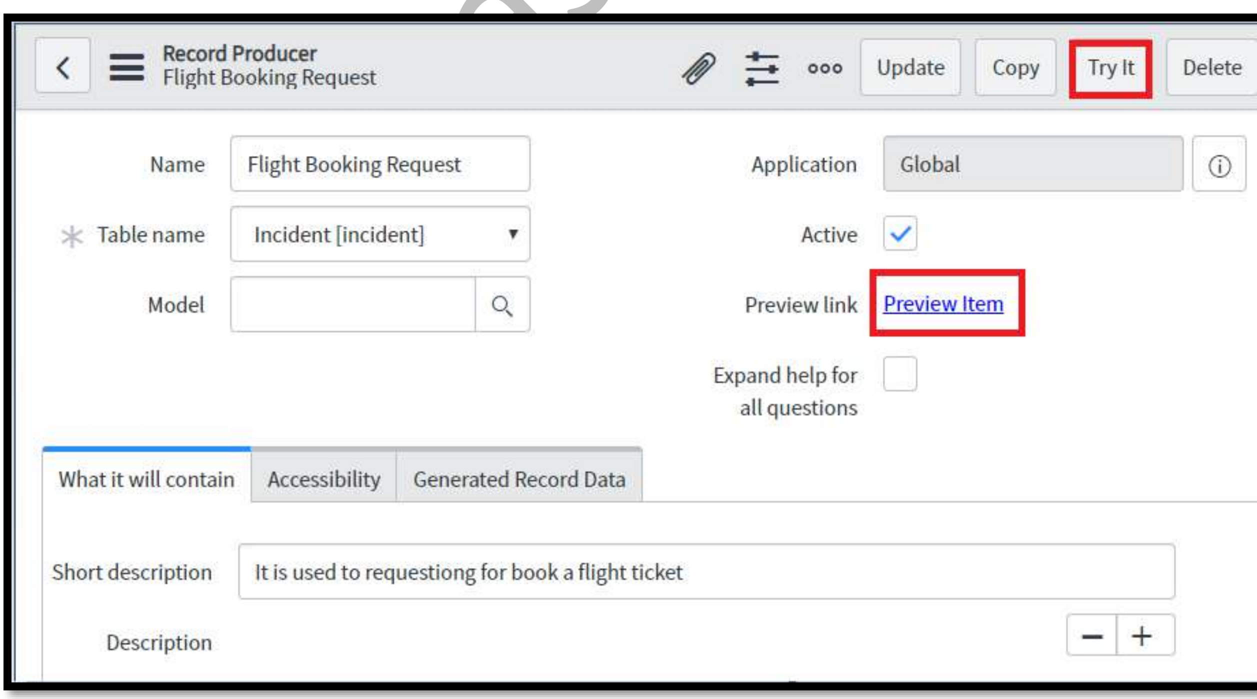




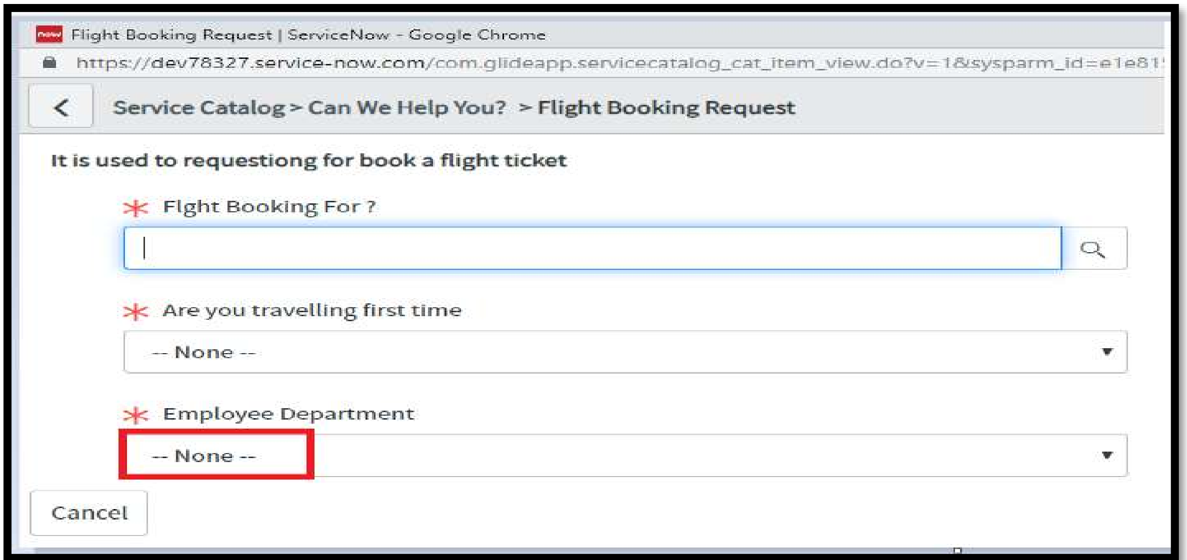
1. **Variable Type**: Lookup Select Box
2. **Mandatory:** Checked True
3. **Order:** 300
4. **Question:** Employee Department



5. Open Type **Specification Tab**
6. **Lookup from table:** User [sys_user]
7. **Lookup value field:** Department
8. **Include none:** Checked **True**
9. **Unique values only:** Checked **True**
10. Click on **Submit**



11. Click on **Preview Item or Try It** to preview the record producer



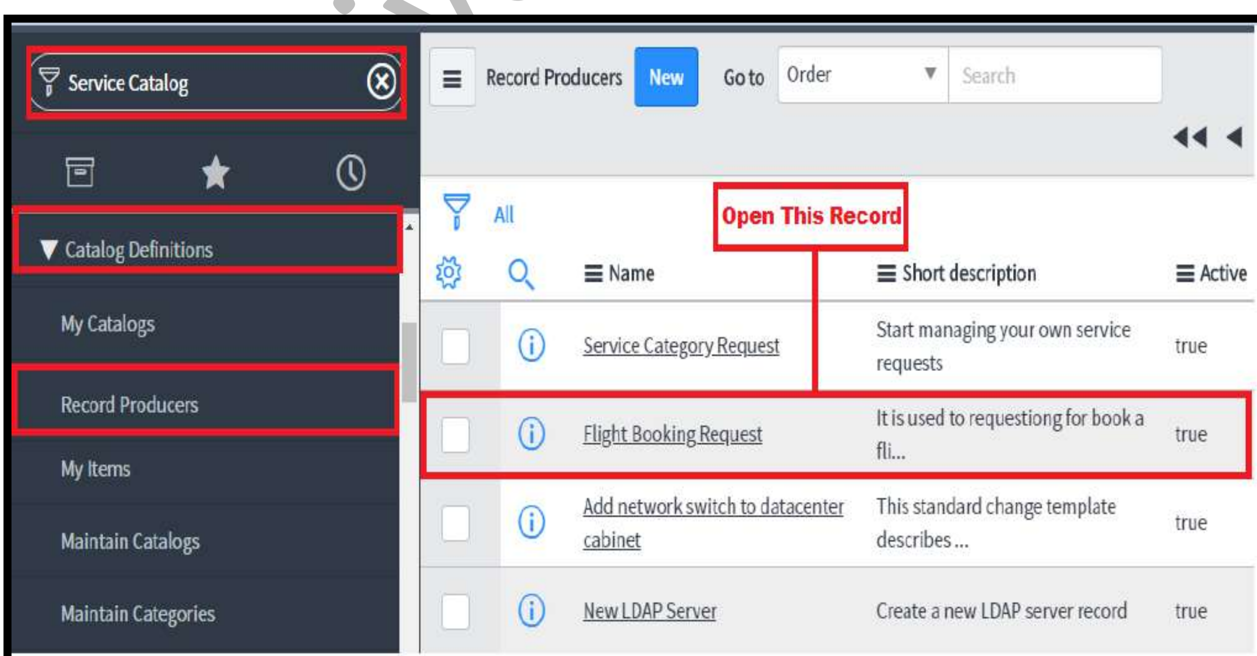
12. Preview of **Record Producer Variable**

Working with Email Variable

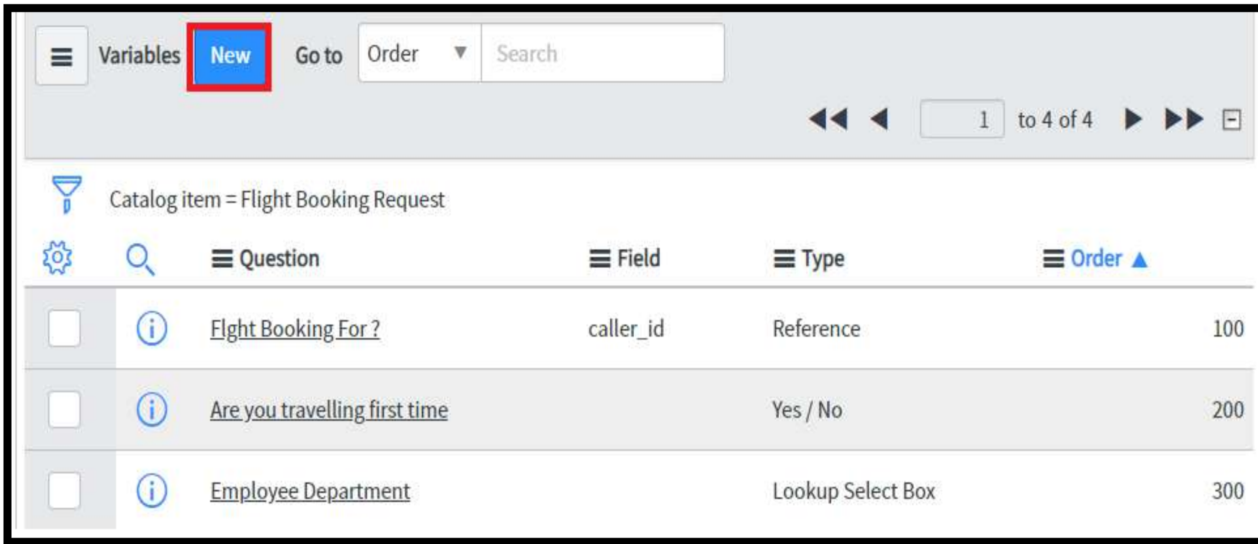
Use to get **Email** id from customer through record producer

Procedure

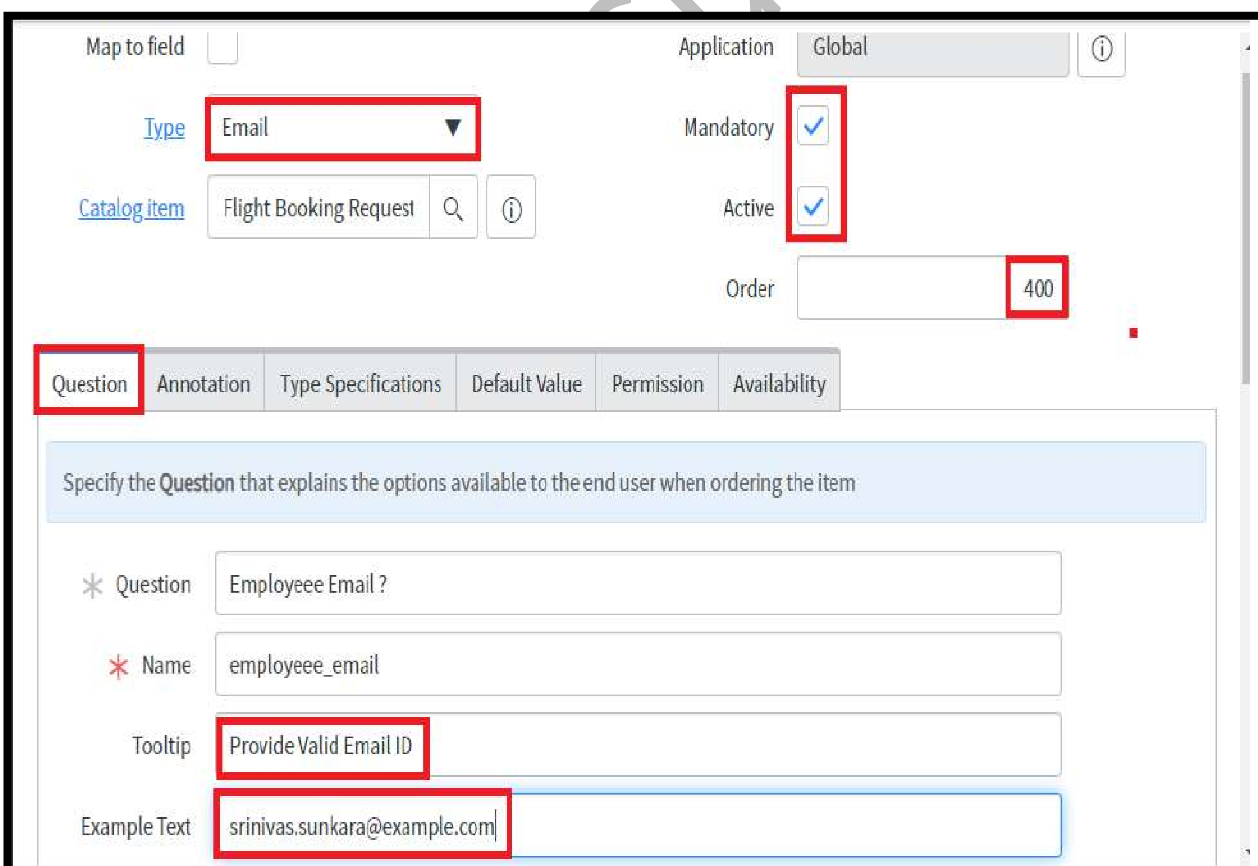
1. Open **Flight Booking Request** Record Producer



2. Open **Variable Tab** from Related List
3. Click on **New**

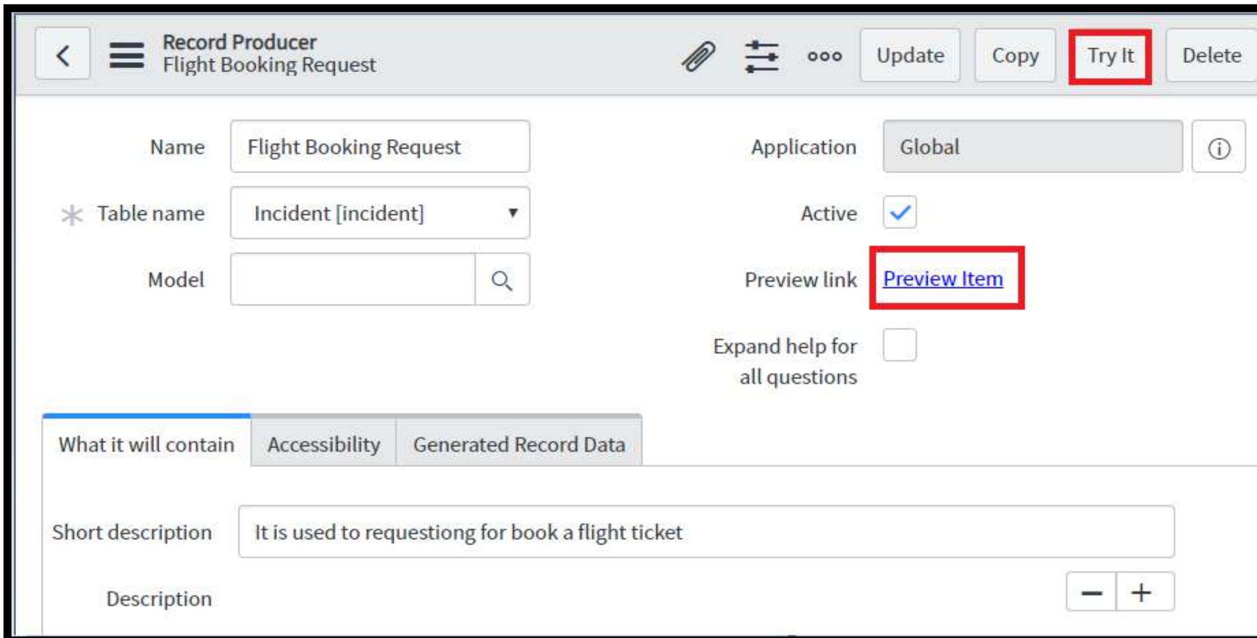


4. Fill the **Email Variable** form

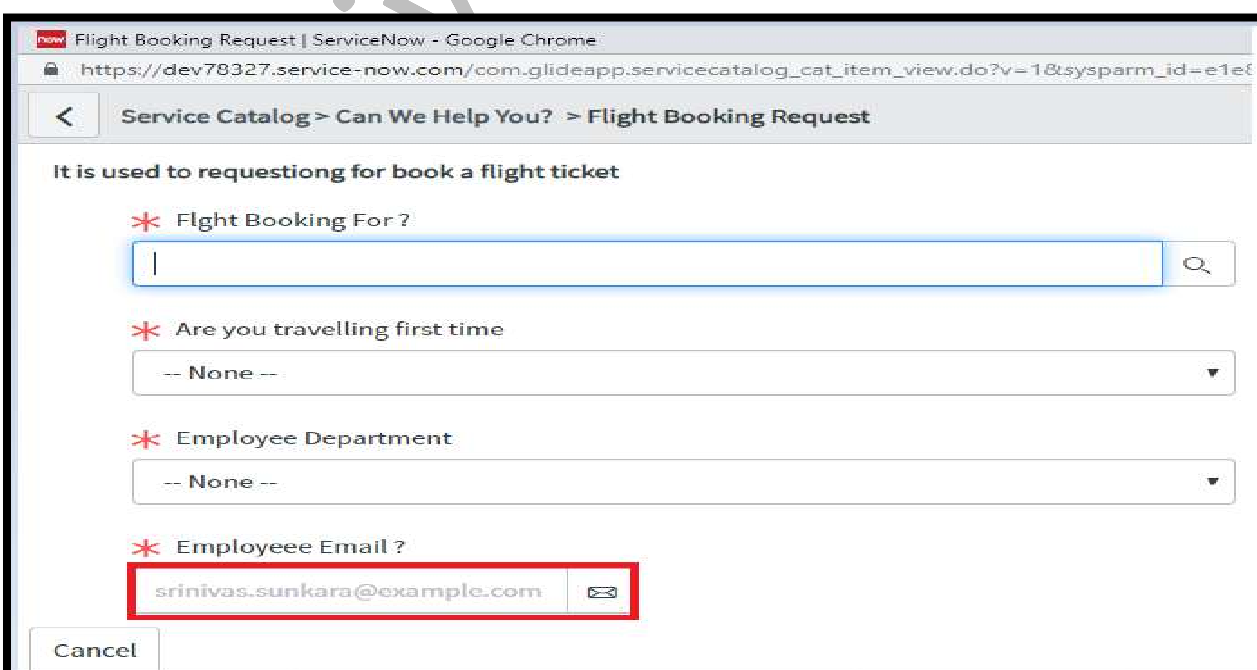


5. **Variable type:** Email

6. **Mandatory:** Checked **True**
7. **Order: 400**
8. **Question:** Employee Email?
9. **Tooltip:** Provide Valid Email ID
10. **Example Text:** srinivas.sunkara@example.com
11. Click on **Submit**



12. Click on **Preview Item or Try It** to preview the record producer



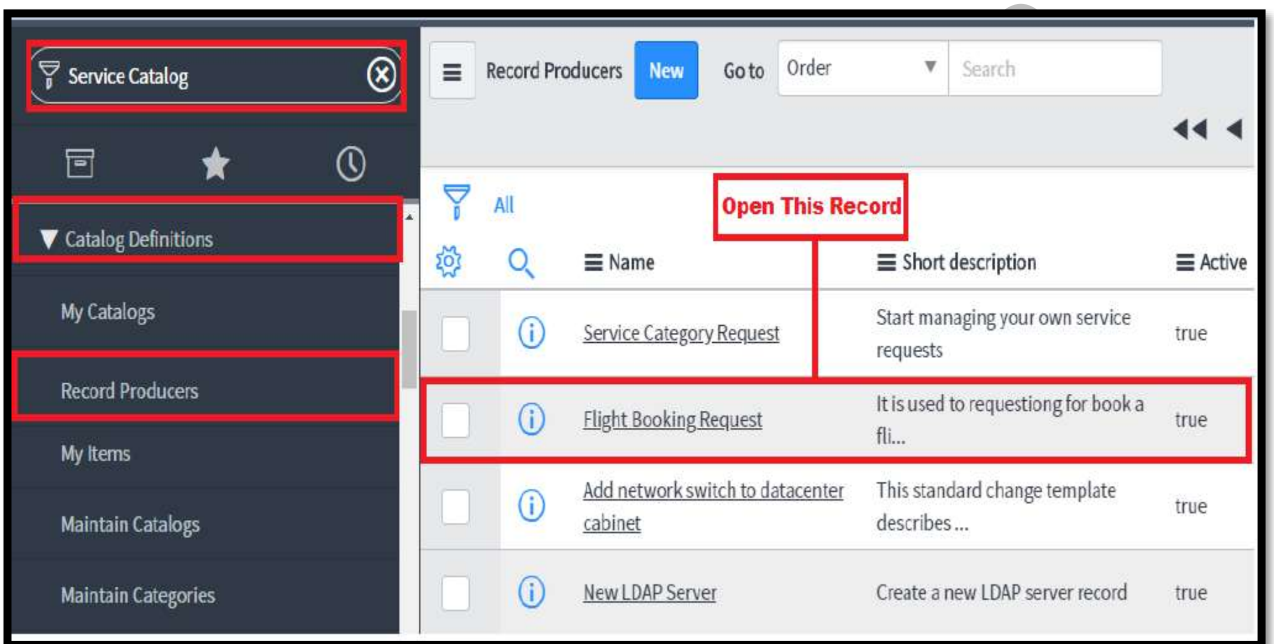
13. Preview of **Record Producer Variable**

Working with Date&Time variable

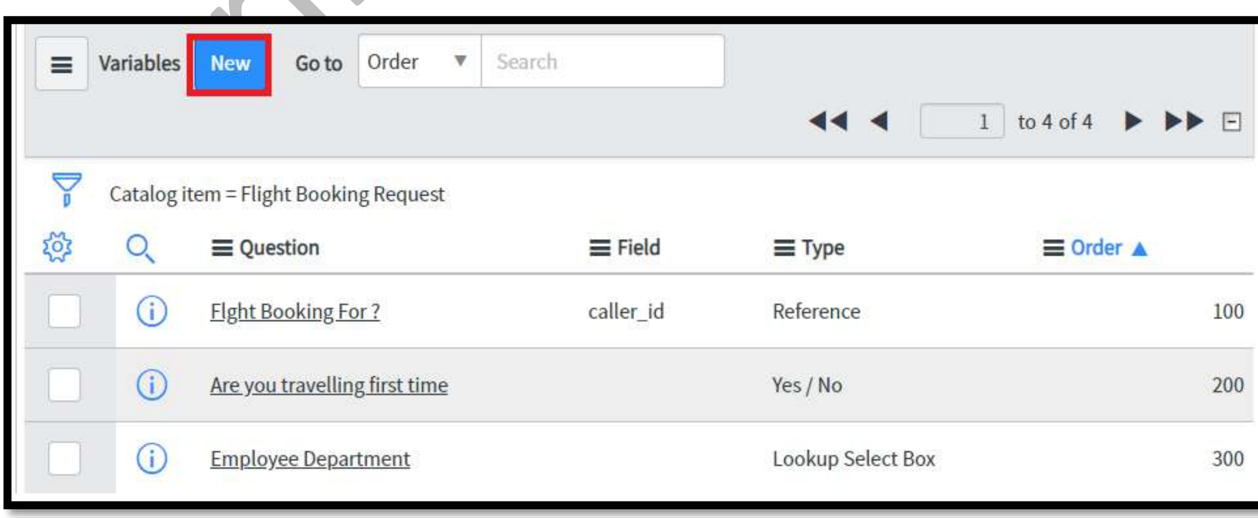
Get date and time to display both **date and time** values. Time value is stored in **Greenwich Mean Time (GMT)** and displayed in the time zone of the current user.

Procedure

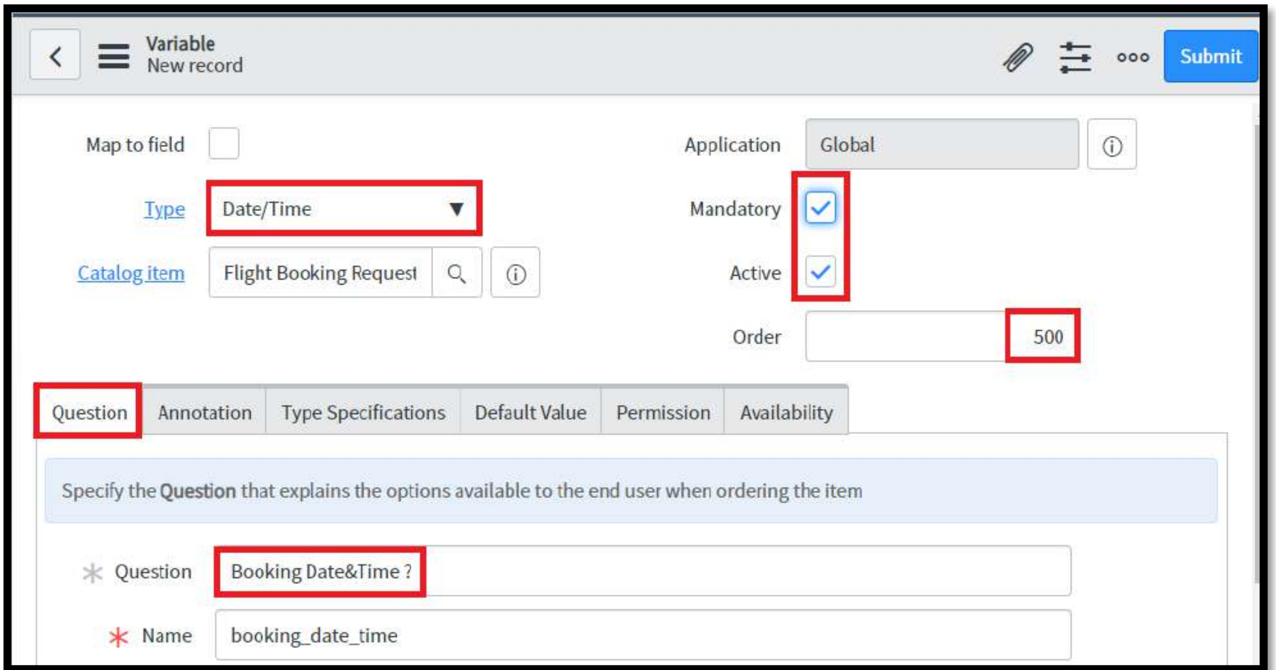
1. Open **Flight Booking Request** Record Producer



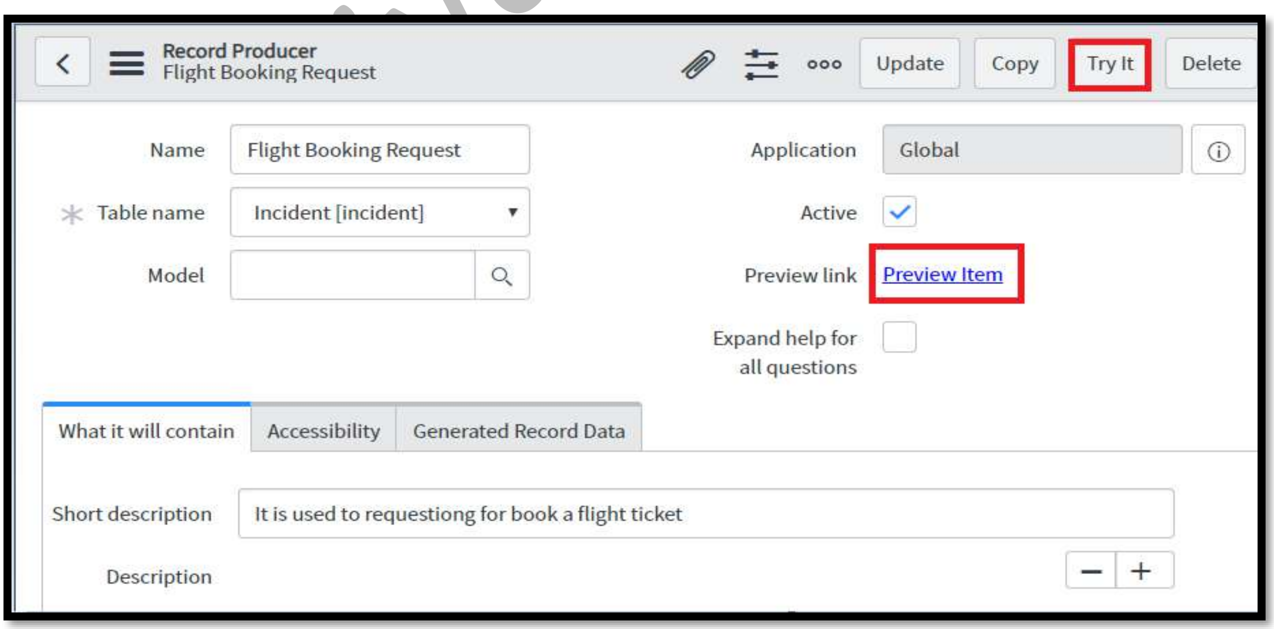
2. Open **Variable Tab** from Related List
3. Click on **New**



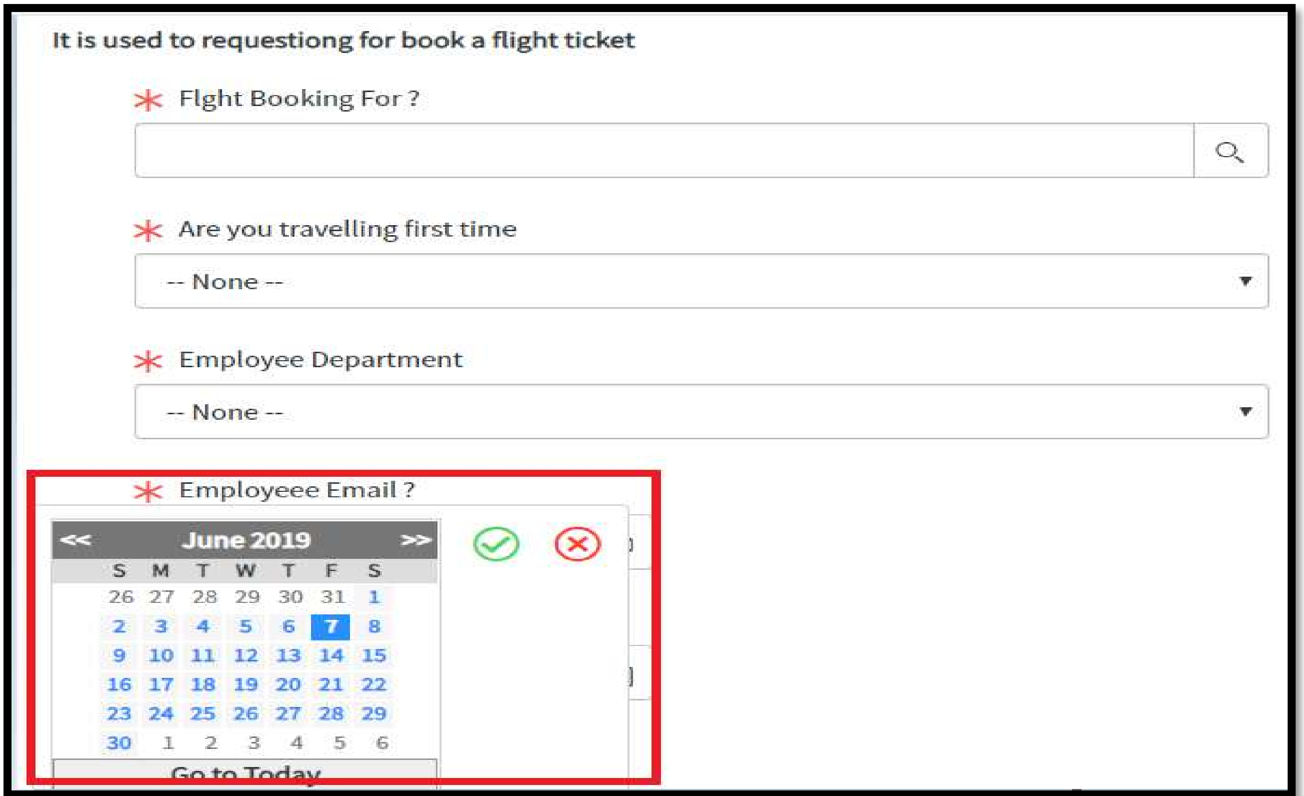
4. Fill the **Date&Time Variable** form



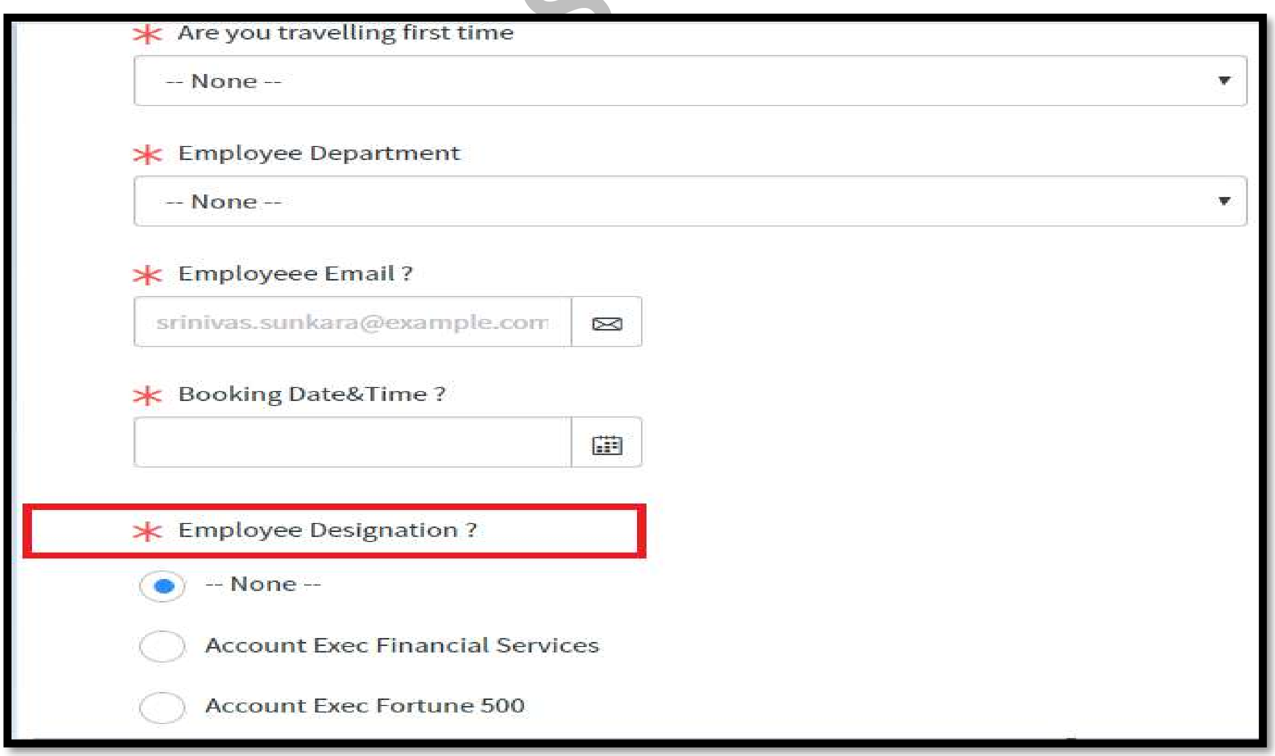
5. **Variable type:** Date/Time
6. **Mandatory**: Checked True
7. **Order**: 500
8. **Question:** Booking Date&Time
9. Click on **Submit**



10. Click on **Preview Item or Try It** to preview the record producer



11. Preview of **Record Producer Variable**

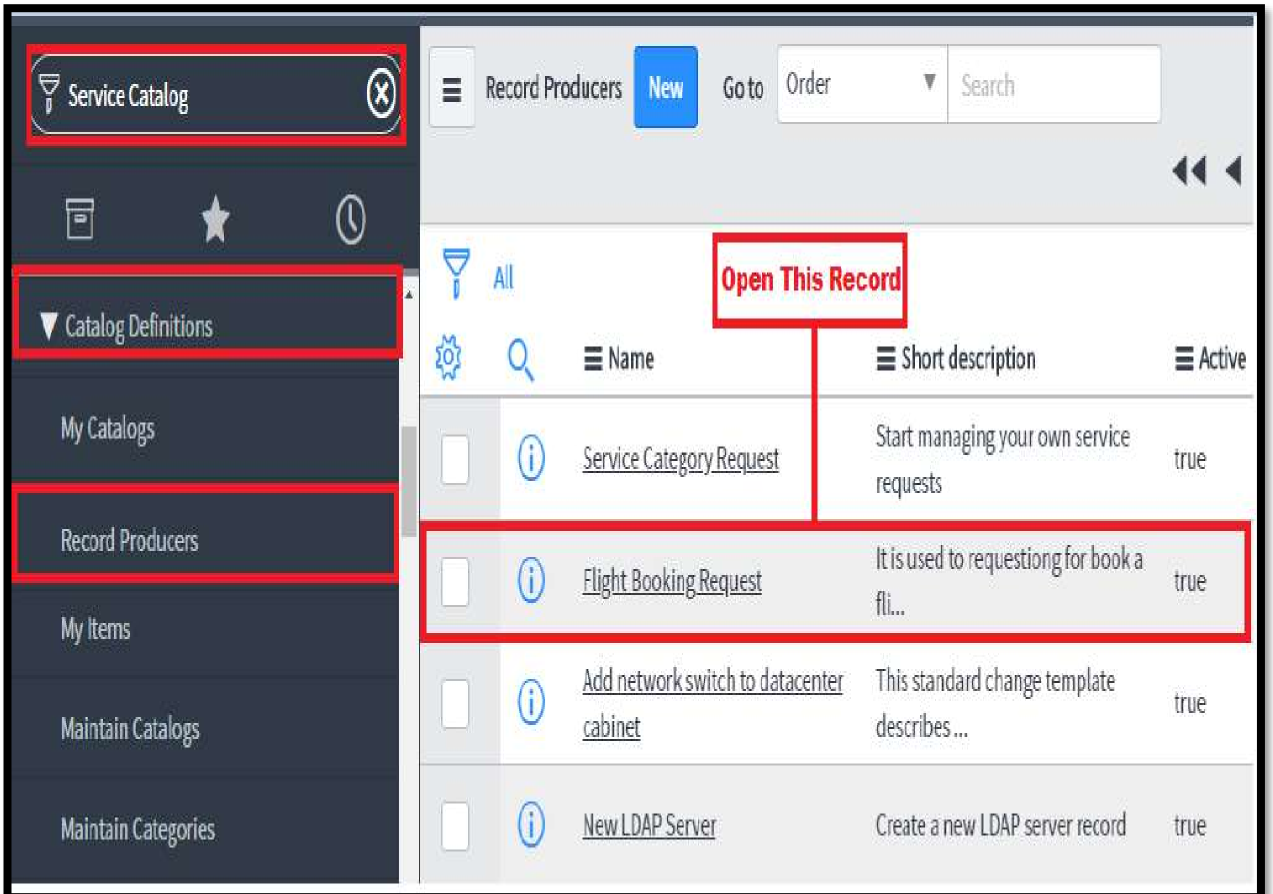


Working with **Lookup Multiple Choice** Variable

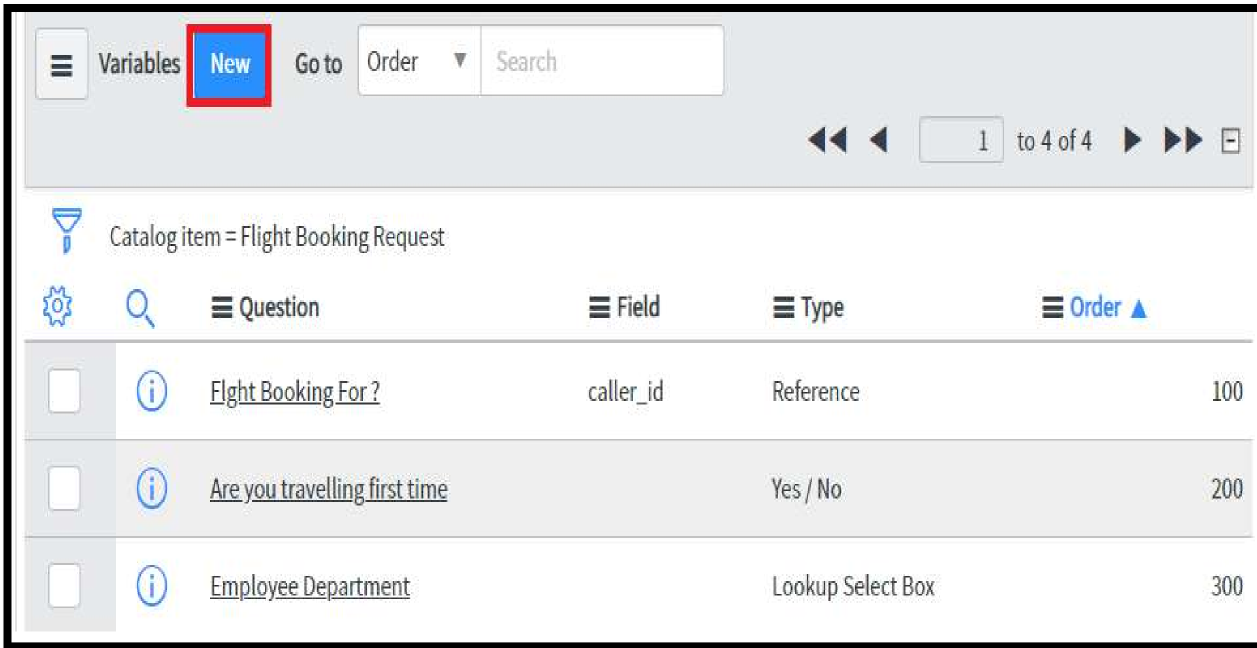
Giving options as radio buttons getting data from a table. Functionality is similar to **Lookup Select Box**, which is used to get choice values from table

Procedure

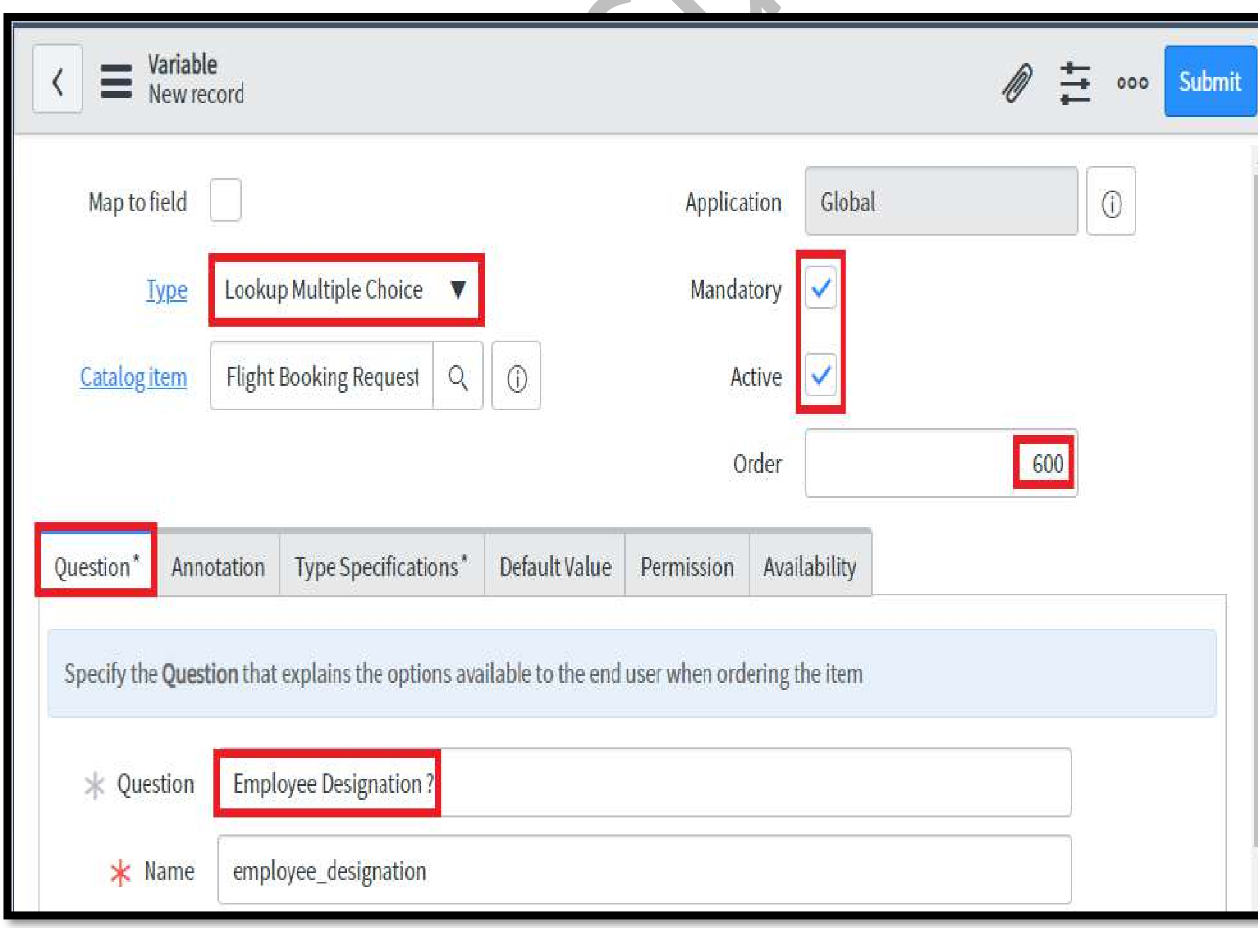
1. Open **Flight Booking Request** Record Producer



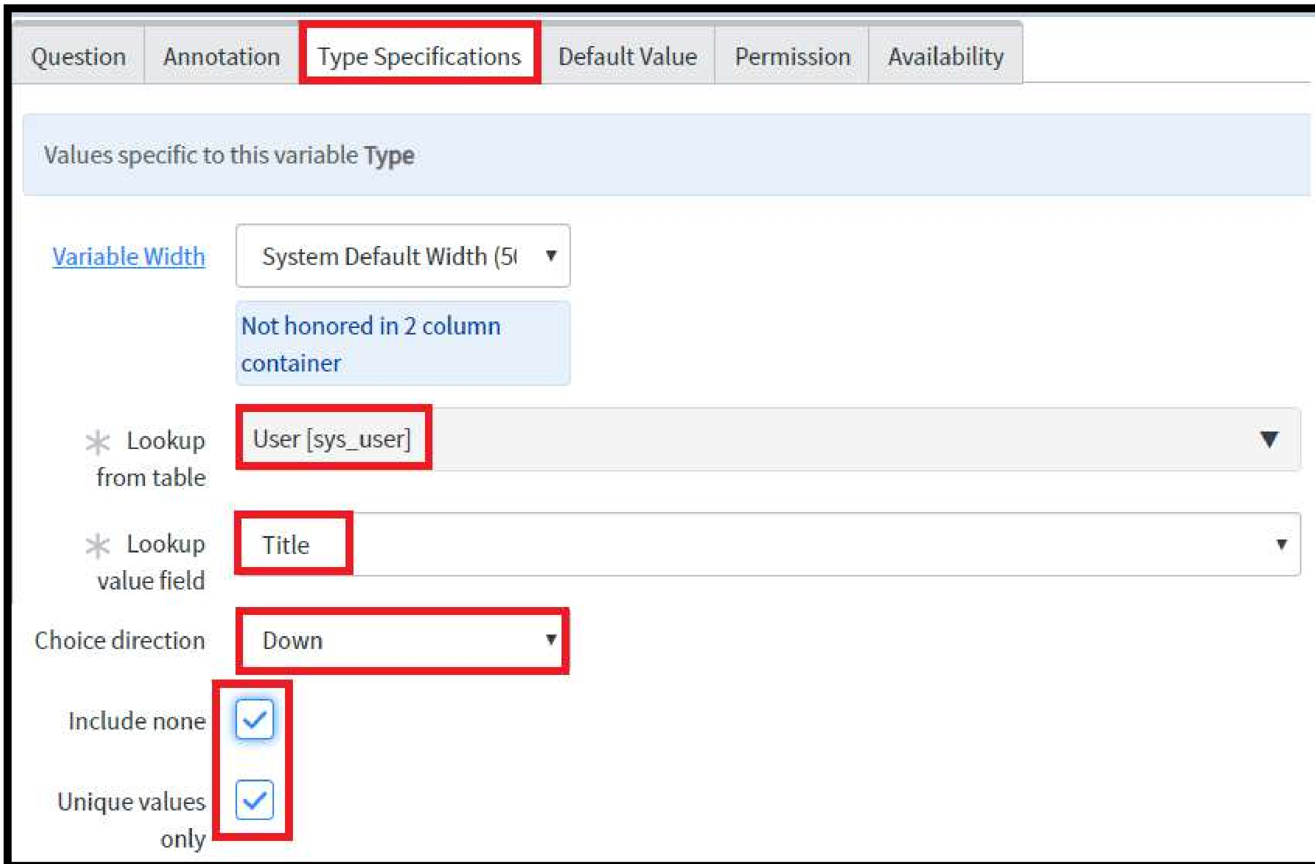
2. Open **Variable Tab** from Related List
3. Click on **New**



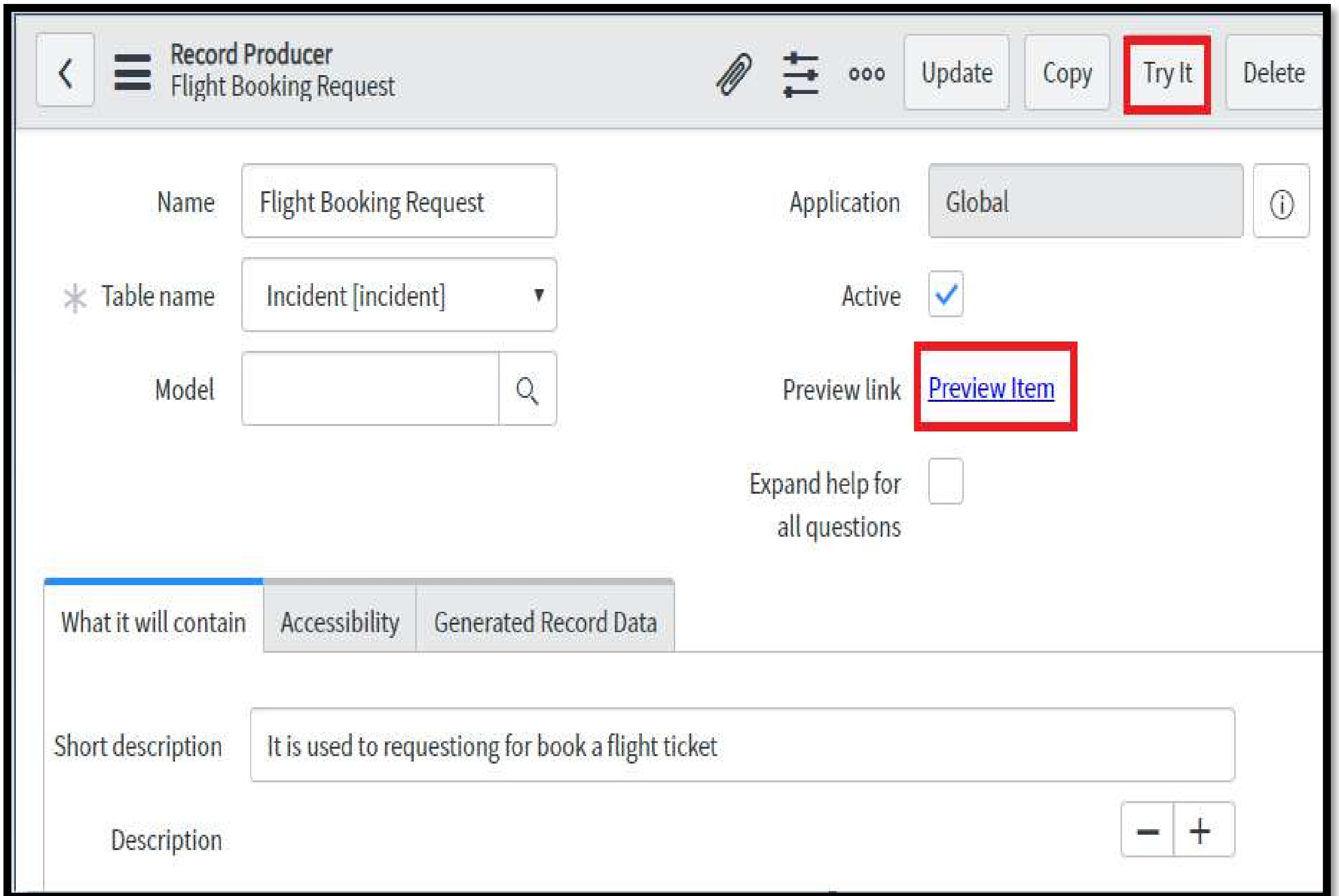
4. Fill the **Lookup Multiple Choice Variable** form



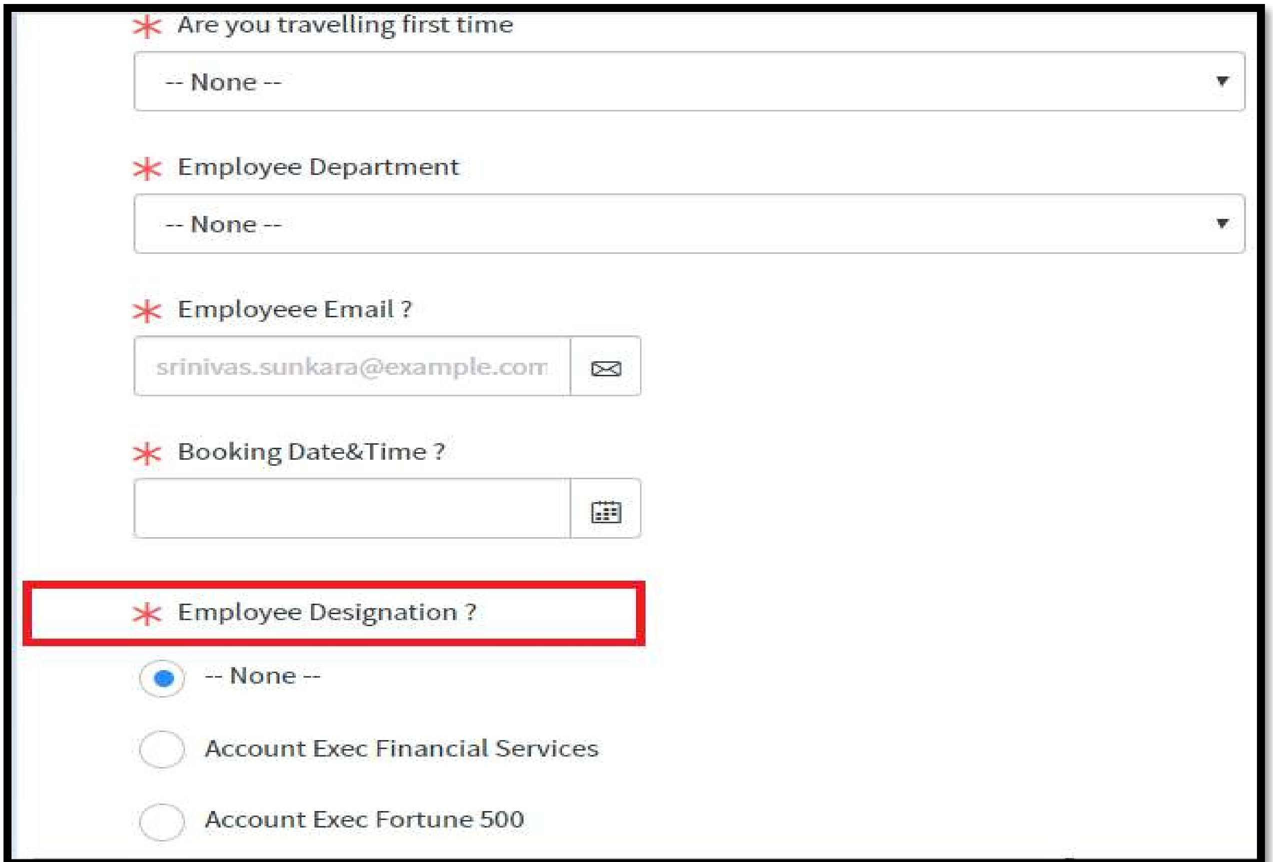
5. **Variable type:** Lookup Multiple Choice
6. **Mandatory**: Checked True
7. **Order**: 600
8. **Question:** Employee Designation?
9. Click on **Submit**



10. Open **Type Specification** tab
11. **Lookup from table:** user[sys_user]
12. **Lookup value field:** Title
13. **Choice direction:** Down
14. **Include none:** Checked **True**
15. **Unique values only:** Checked **True**
16. Click on **Submit**



16. Click on **Preview Item or Try It** to preview the record producer



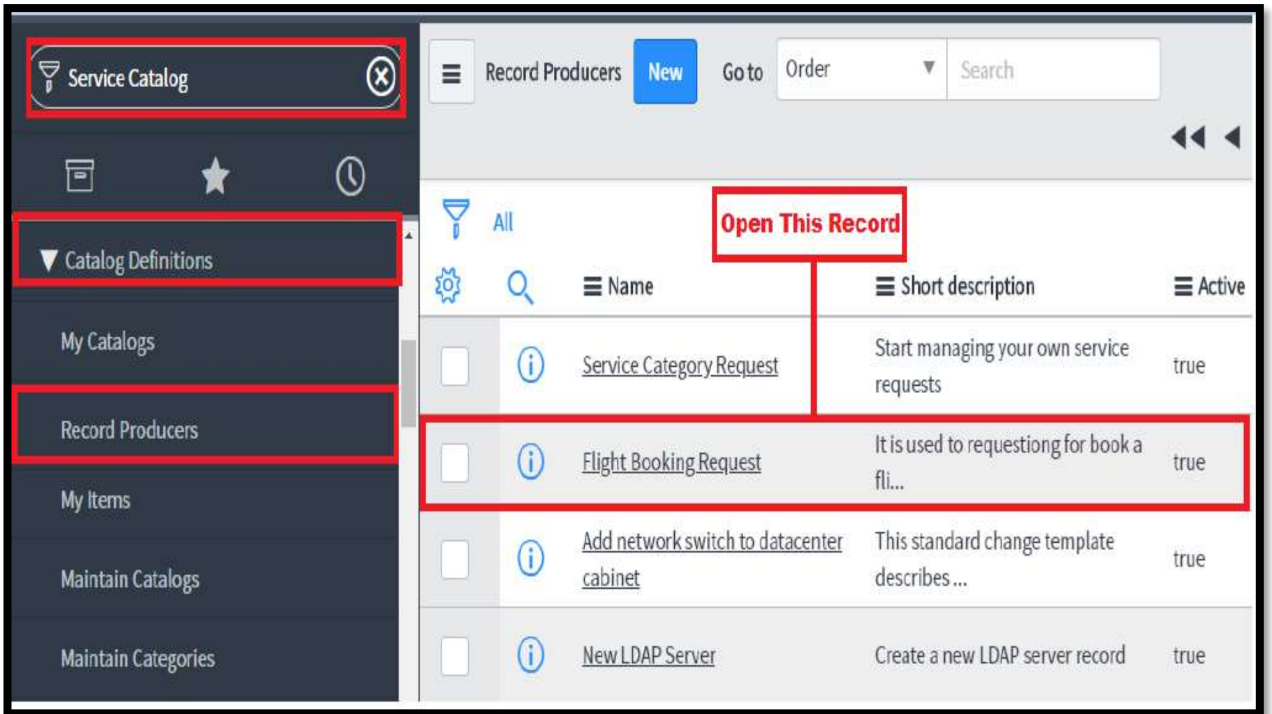
Working with **Break** Variable

A **Break Variable** can display a horizontal line on the form. Breaks are formatting elements. It's not collecting any data from customer

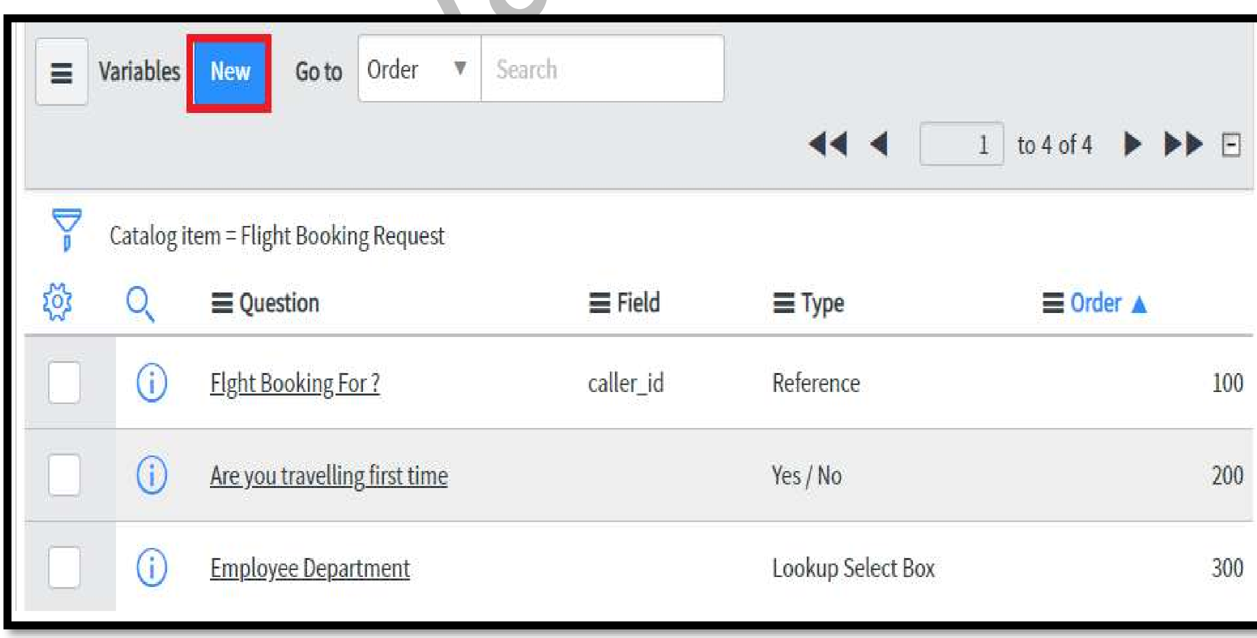
Note: The break variable is not yet supported on **Service Portal** and **Mobile Devices.**

Procedure

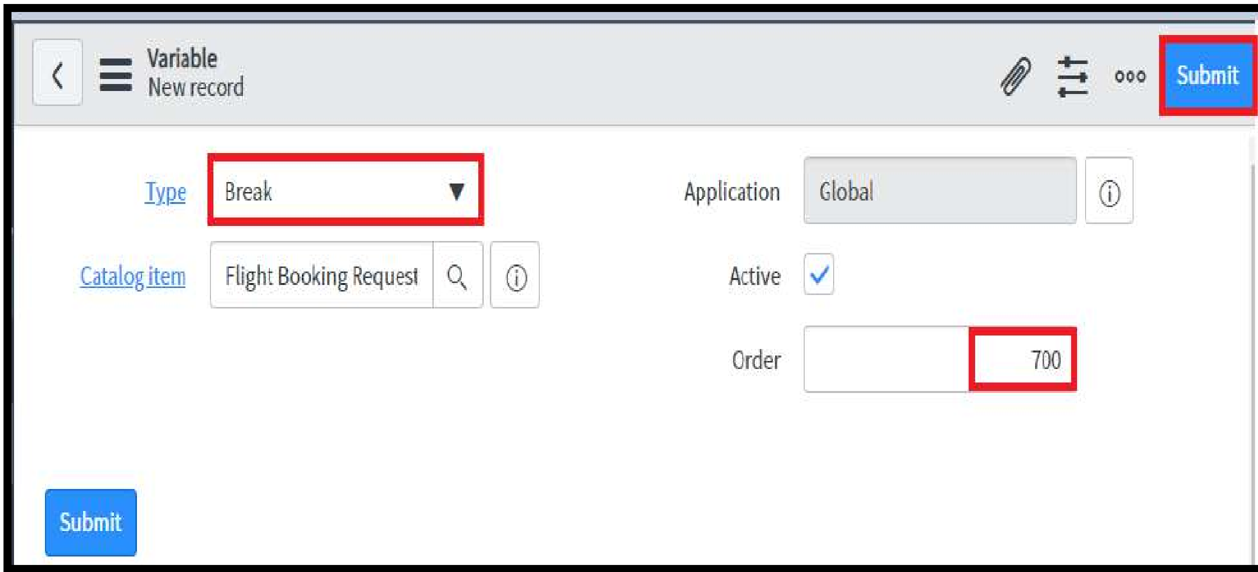
1. Open **Flight Booking Request** Record Producer



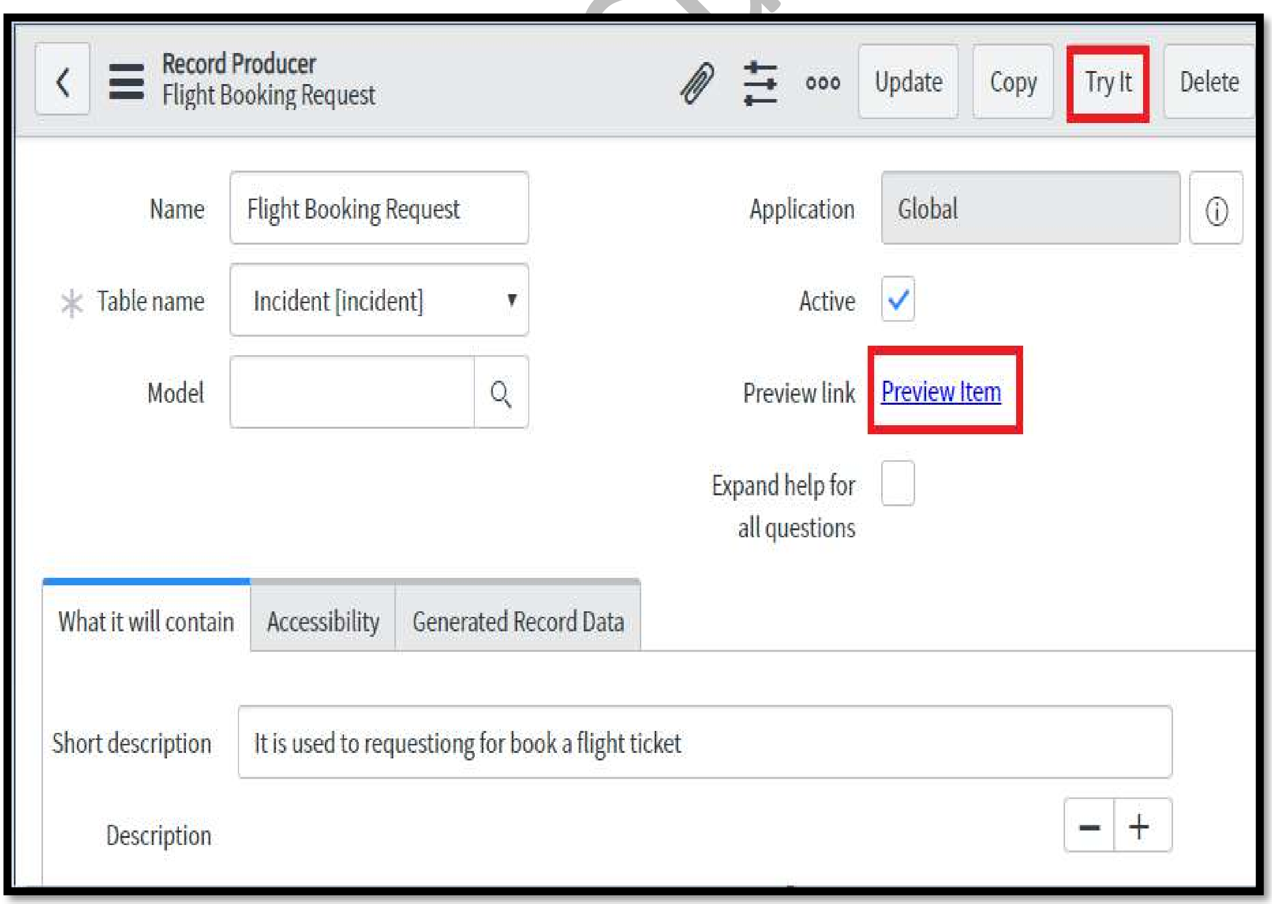
2. Open **Variable Tab** from Related List
3. Click on **New**



4. Fill the **Break Variable form**



5. **Variable Type:** Break
6. **Order:** 700
7. Click on **Submit**



8. Click on **Preview Item or Try It** to preview the record producer

9. Preview of **Record Producer Variable**

What is Order Guide

1. Order guide is submit a single service catalog request that can be generate bulk of items.
2. Order guides allow the customers to answer a series of questions
3. Resulting in one or more products or services being requested
4. This will avoid the need for customers to know specifically what products and services they required
5. Ensure that costumers request the right items leads to improved service provider efficiency and greater customer satisfaction
6. Order guides can be defining which catalog items to order by evaluating order guide rule conditions.
7. Information the customer provide within the order guide can be passed as cascading variables to the ordered items, allowing common information to be reused across multiple items.

Ex: **New On-boarding** order guide can contain number of items that new employees commonly need, such as **Computer, ID Cards, Mobile Phone, New mail creation, Other facilities**. After choose this particular order guide, the employee can then provide required information about the new employee, including **Designation** and **Department.** The order guide then submits an order for catalog items like computer, based on the details provided.

All Order guides can be run automatically, generating a set of ordered items without needing to manually submit a service catalog request.

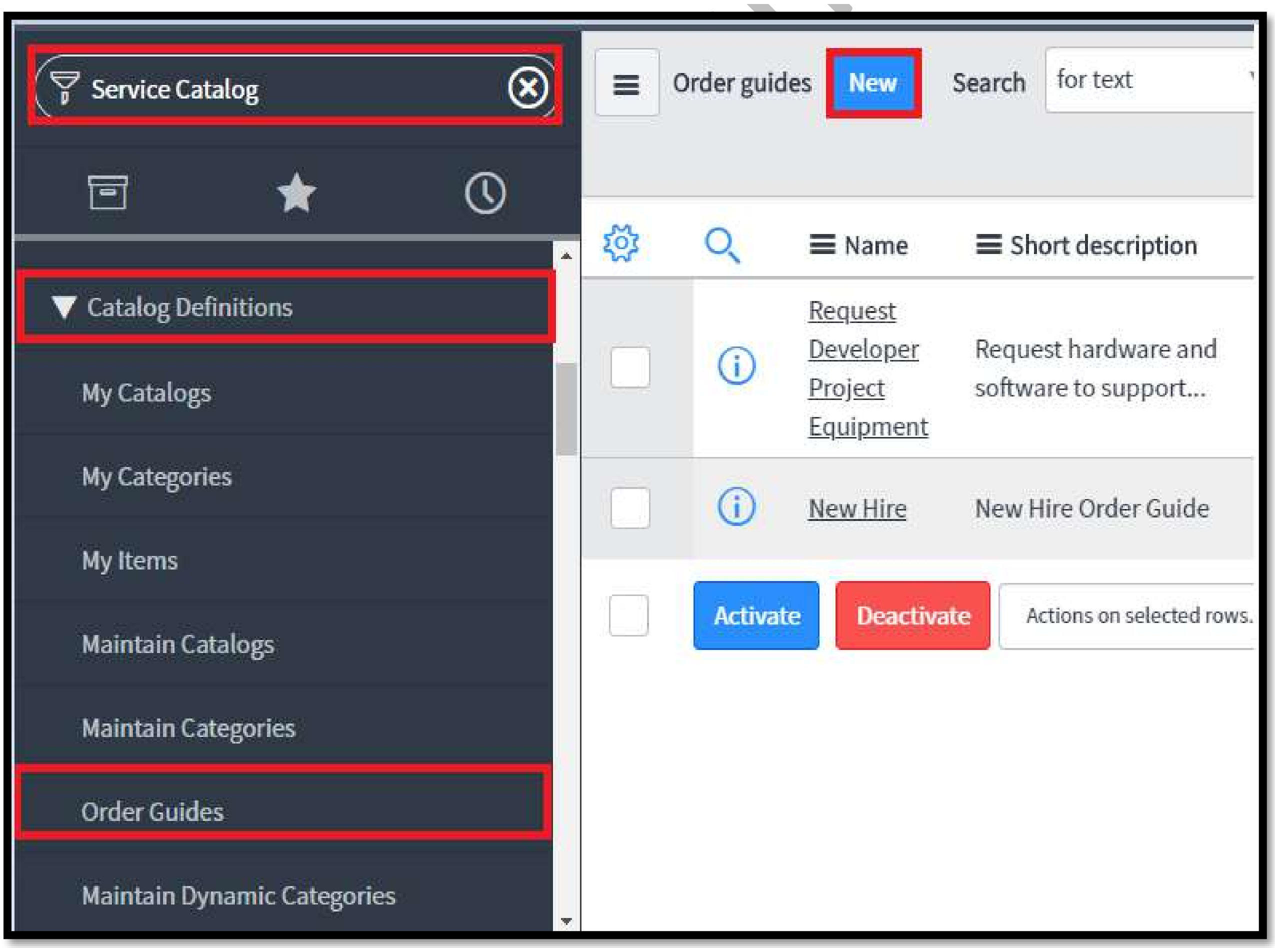
For example, an **on-boarding** workflow for a new employee can automatically run an order guide to order items for that employee.

Exercise: 7 Create On-Boarding Order Guide

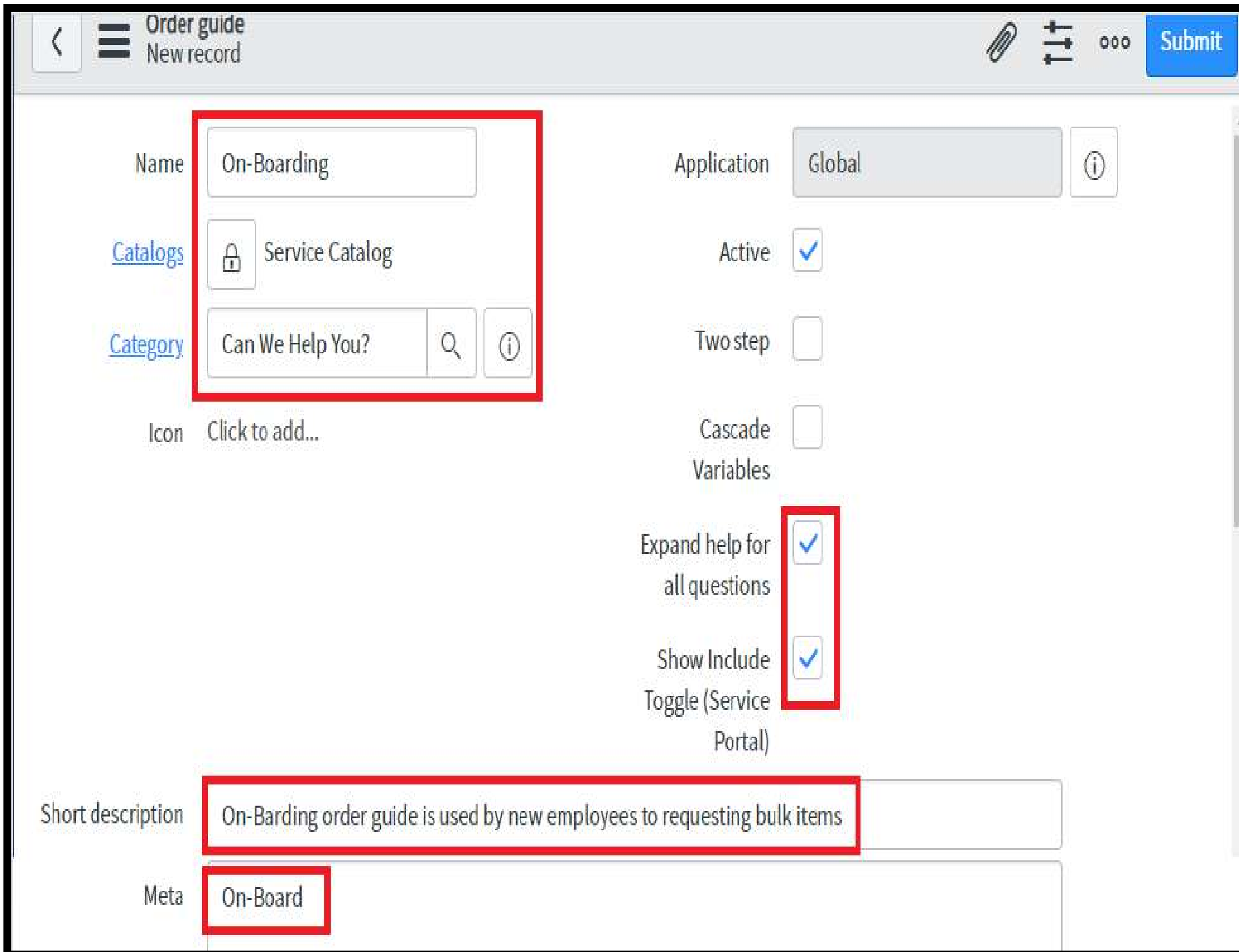
We can create an order guide for new employees to requesting bulk of items in single request, needed for business purpose

Procedure

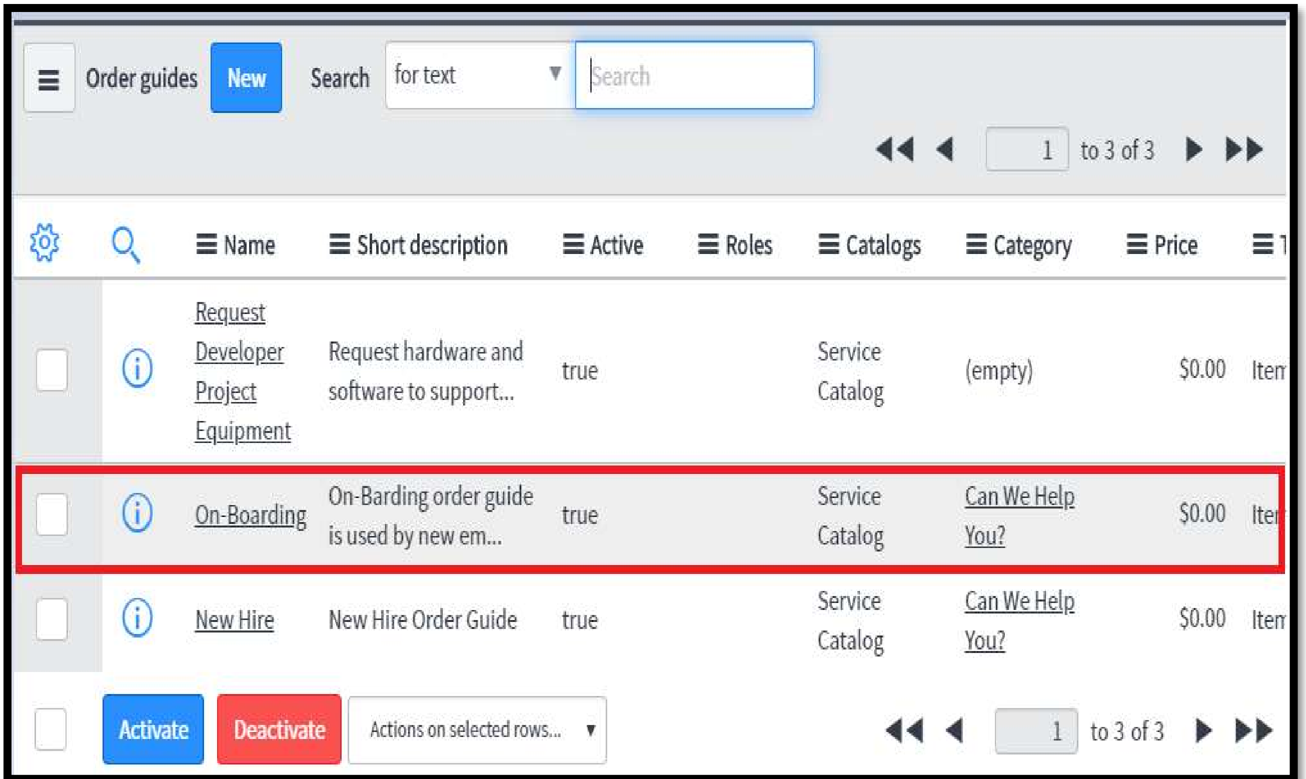
1. Navigate to **Service Catalog** > **Catalog Definition** > **Order Guides**.
2. Click on **New**



3. Fill **On-Boarding** order guide form



4. **Name:** On-Boarding
5. **Catalog:** Service Catalog
6. **Category:** Can we help you?
7. **Icon:** (Optional if required add image)
8. **Expand help for all Questions:** Checked **True**
9. **Show Include Toggle** (Service Portal): Checked **True**
10. **Short Description:** On-Barding order guide is used by new employees to requesting bulk items
11. **Meta:** On-Board
12. Click on **Submit**



13. Sucessfully created new **Order Guide** for look into screen

Working with Container Start Variable

Container Start Variable is Similar to Variable Set. This variable can define a layout for a container that can hold multiple vraibales to display specific information realated to your business

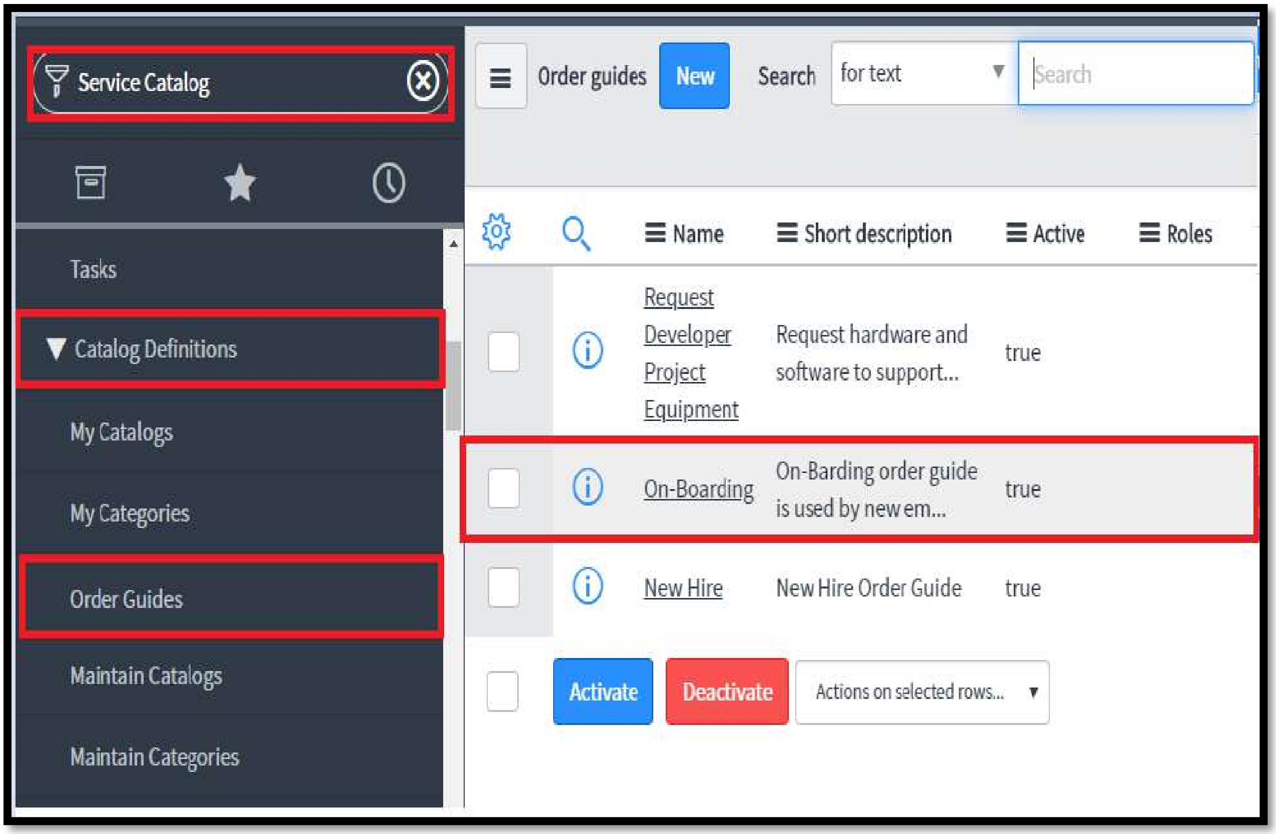
These variables use **Container Start** and **Container End** variables to define the start and end points of a container layout.

The container end should be used along with container start to close a container layout.

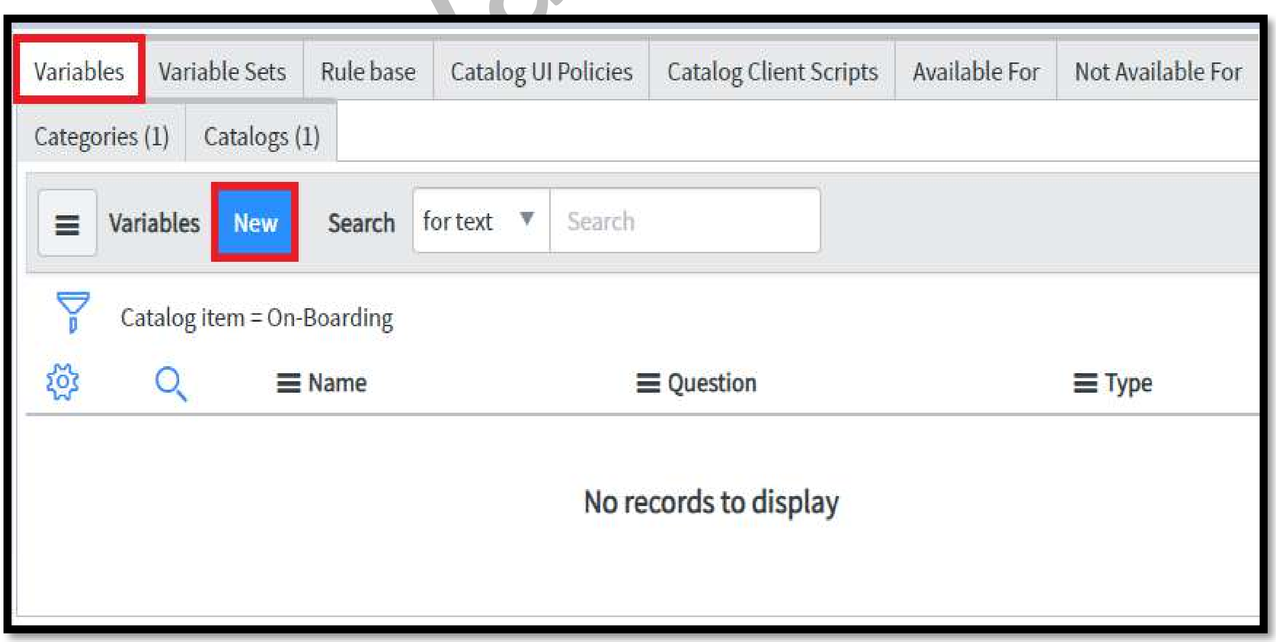
A container layout can be split into two or three columns using the container split variable

Procedure

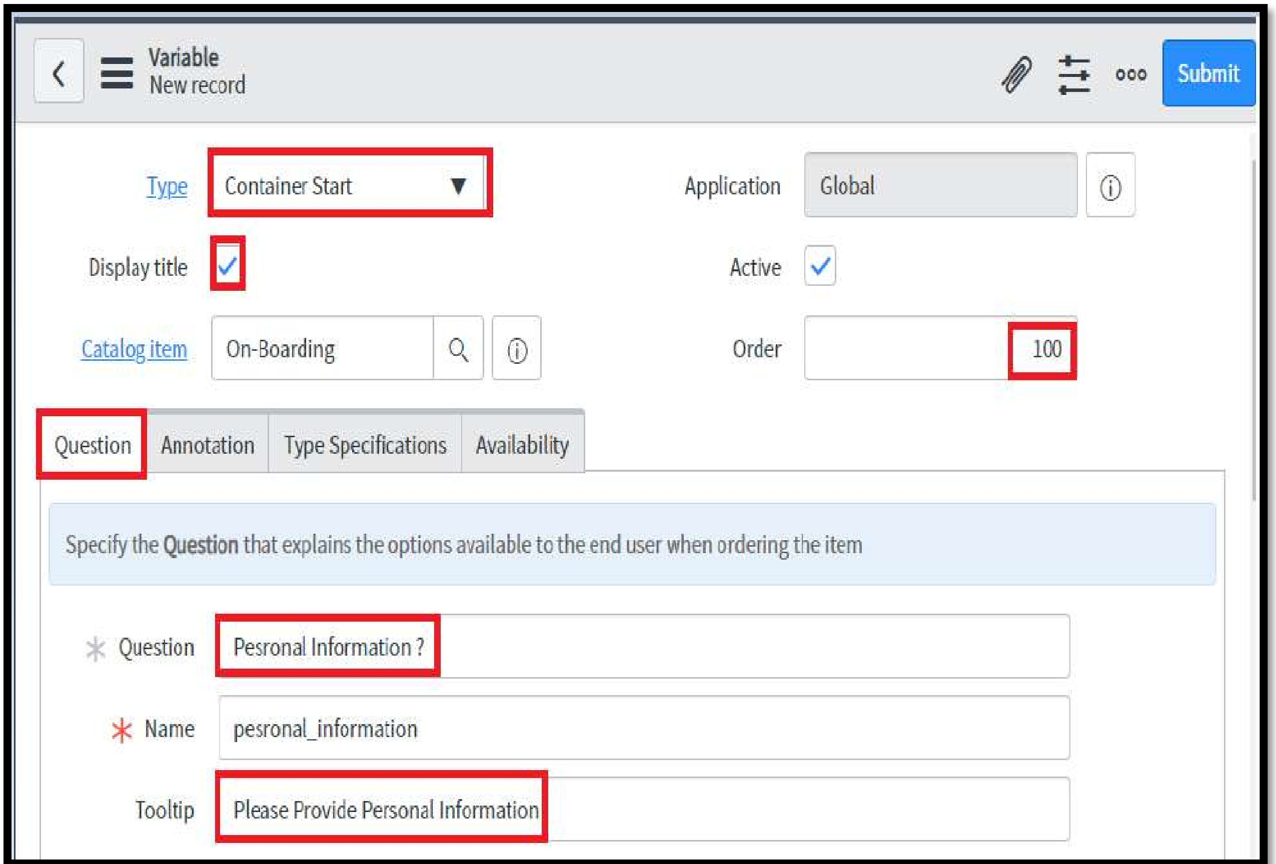
1. Navigate to **Service Catalog** > **Catalog Definition** > **Order Guides**
2. Open **On-Boarding** order guide Record



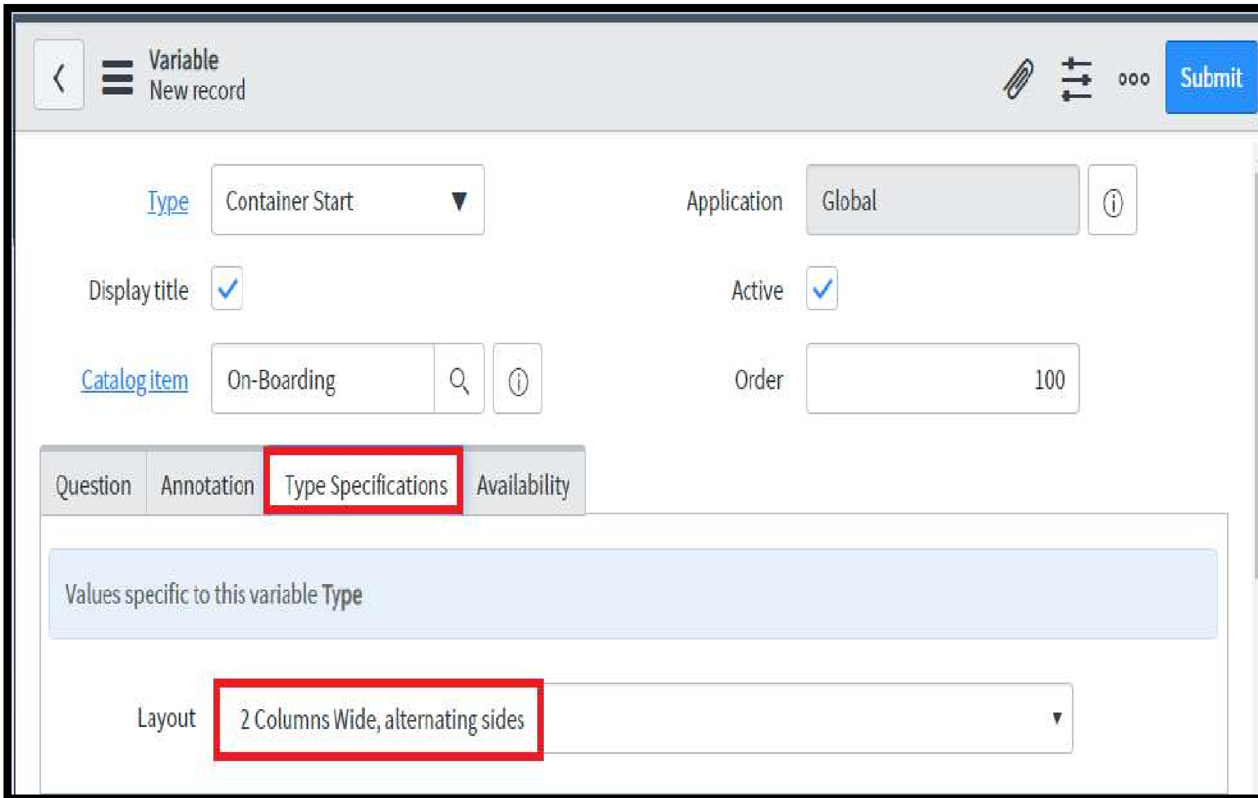
3. Open **Variable Tab** from Related List
4. Click on **New**



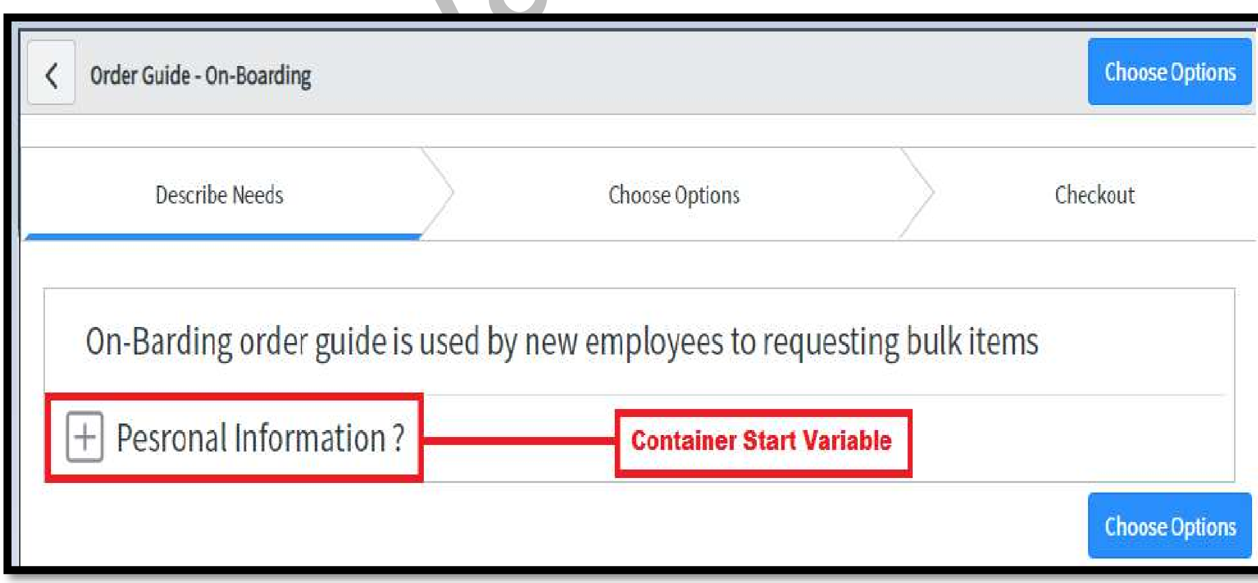
5. Fill **Container Start Variable** Form



6. **Variable Type:** Container Start
7. **Display Title:** Checked **True**
8. **Catalog Item:** On-Boarding
9. **Order:** 100
10. **Question:** Personal Information
11. **Name:** Auto Filled
12. **Tooltip**: Please Provide Personal Information



13. Open **Type Specification** Tab
14. Layout: 2 Columns Wide, alternating sides
15. Click on **Submit**



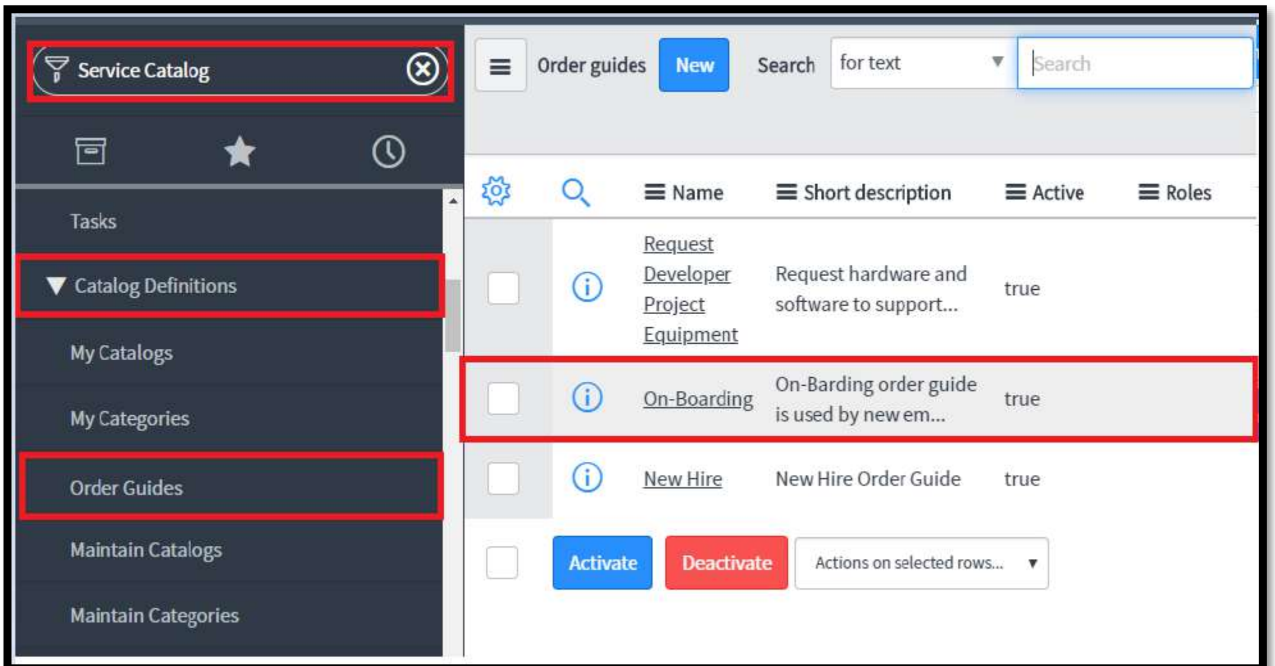
16. Click on **Try It** to preview the **Order Guide**

Working with Single Line Text Variable

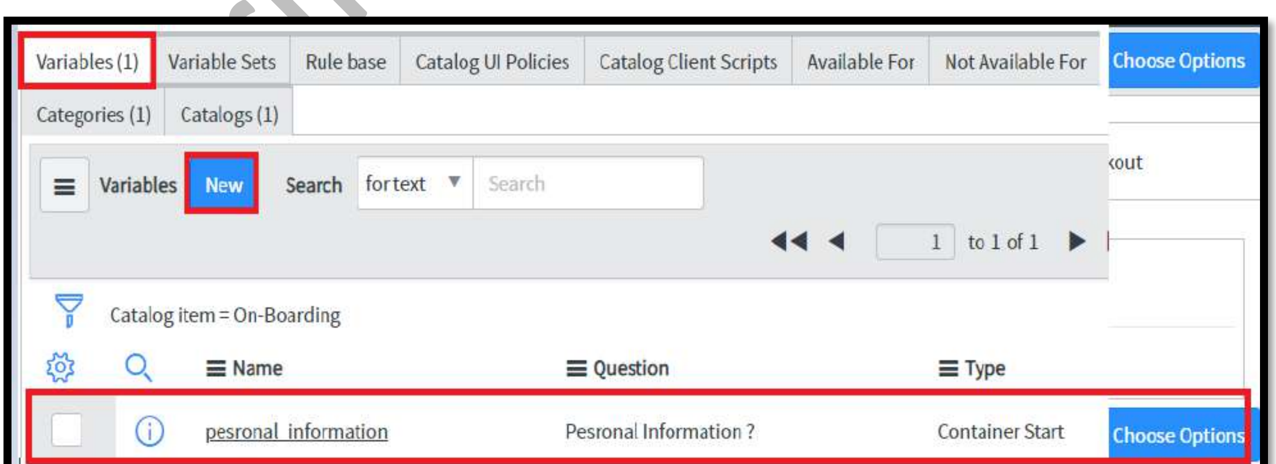
This is used to get single line text value getting from customer

Procedure

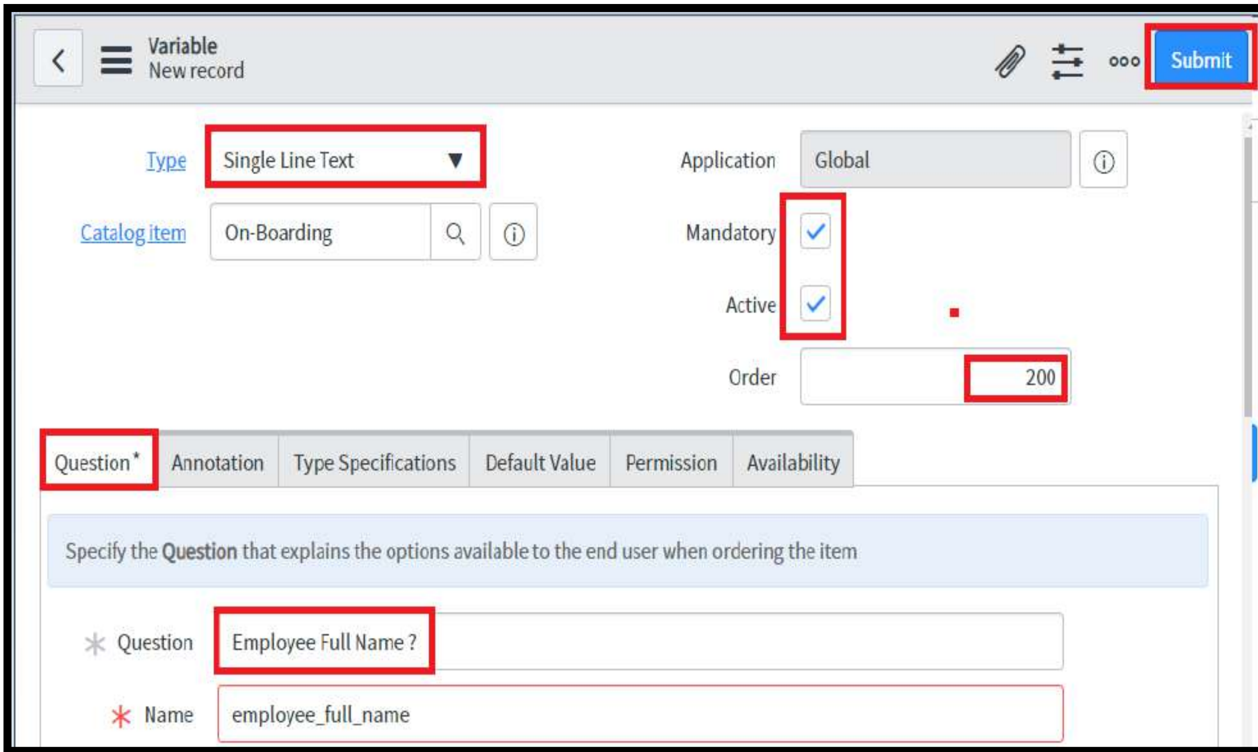
1. Navigate to **Service Catalog** > **Catalog Definition** > **Order Guides**
2. Open **On-Boarding** order guide Record



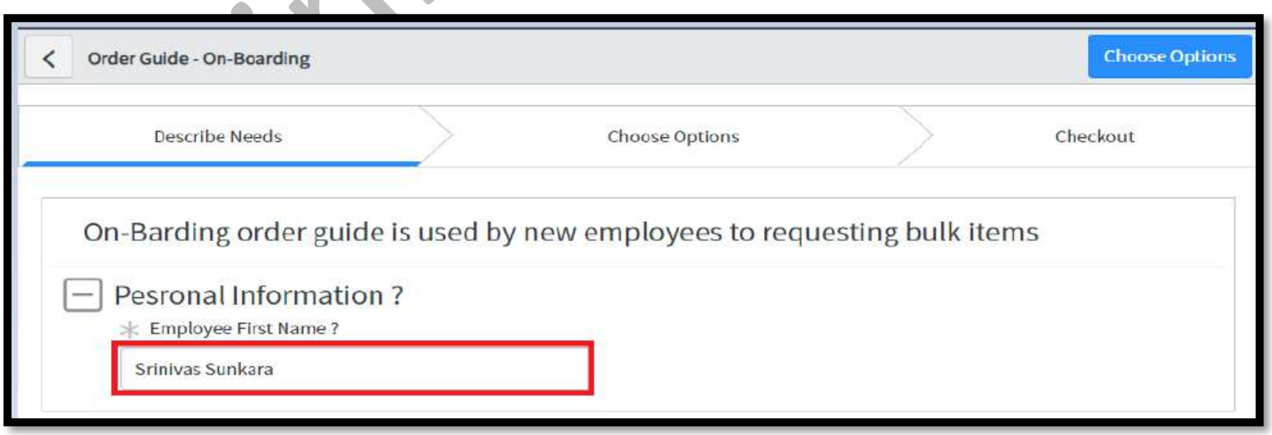
3. Open **Variable Tab** from Related List
4. Click on **New**



5. Fill **Single Line Text** variable form



6. **Variable Type:** Single Line Text
7. **Catalog Item:** On-Boarding
8. **Mandatory:** Checked **True**
9. **Order:** 200
10. **Question:** Employee Full Name?
11. **Name:** Auto Filled
12. Click on **Submit**



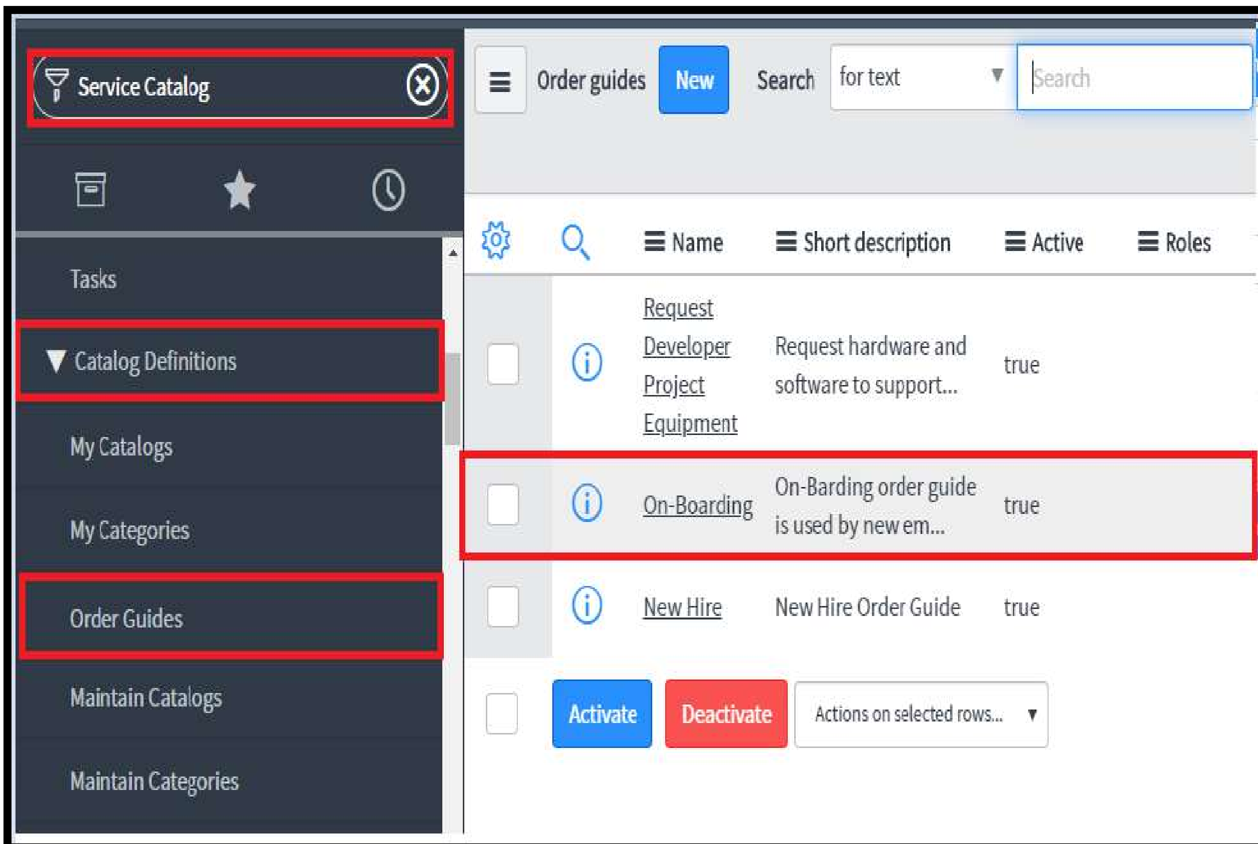
13. Click on **Try It** to preview the **Order Guide**

Working with List Collector Variable Variable

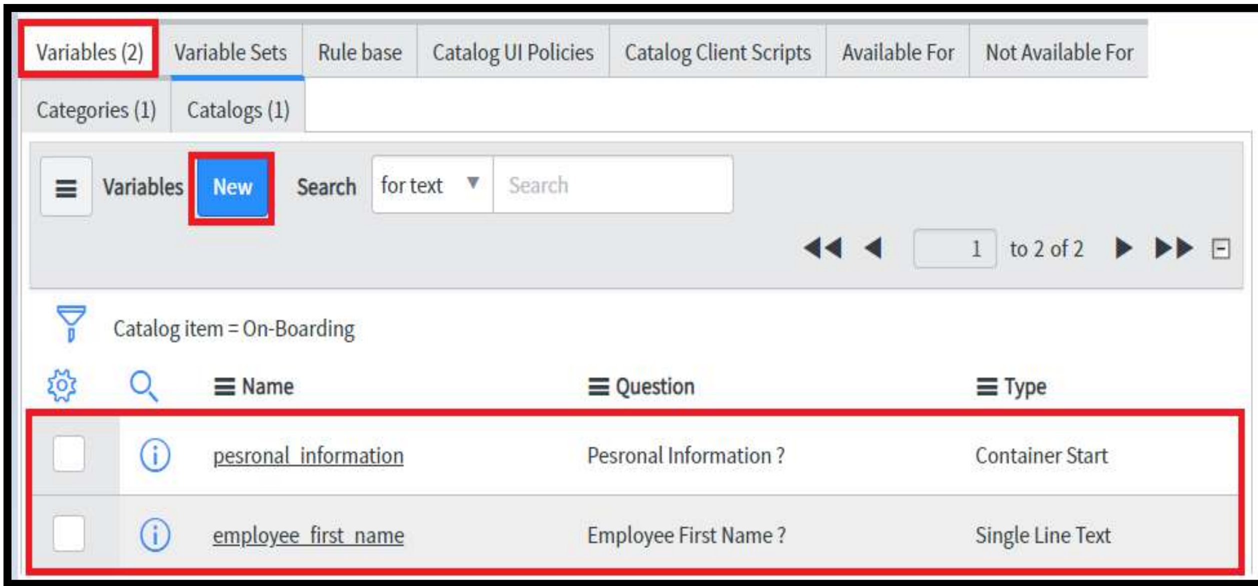
List collector variable is similar to **Reference Variable** used to get reference records from other tables as **Slash Bucket** format, including **Condition Builder**

Procedure

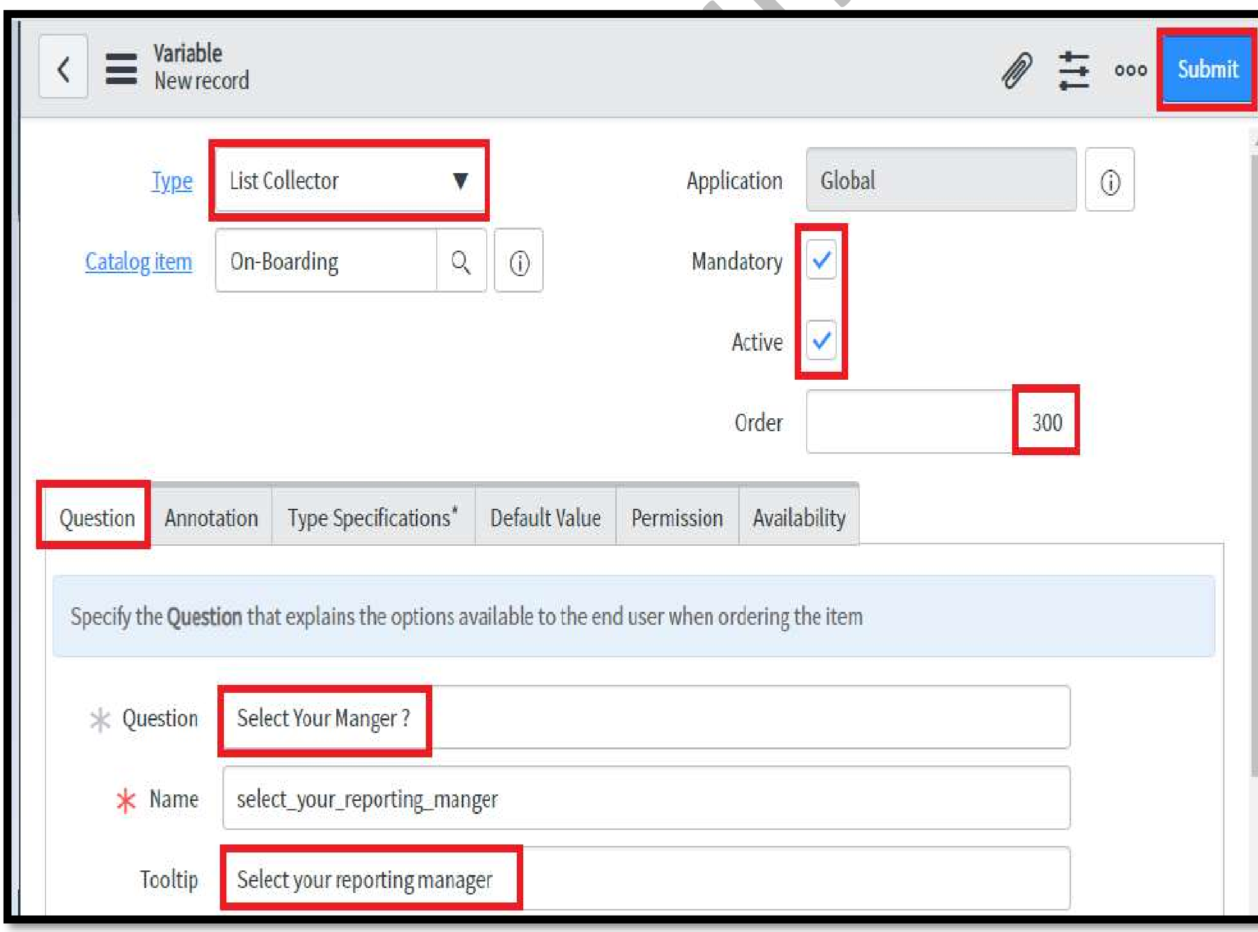
1. Navigate to **Service Catalog** > **Catalog Definition** > **Order Guides**
2. Open **On-Boarding** order guide Record



3. Open **Variable Tab** from Related List
4. Click on **New**

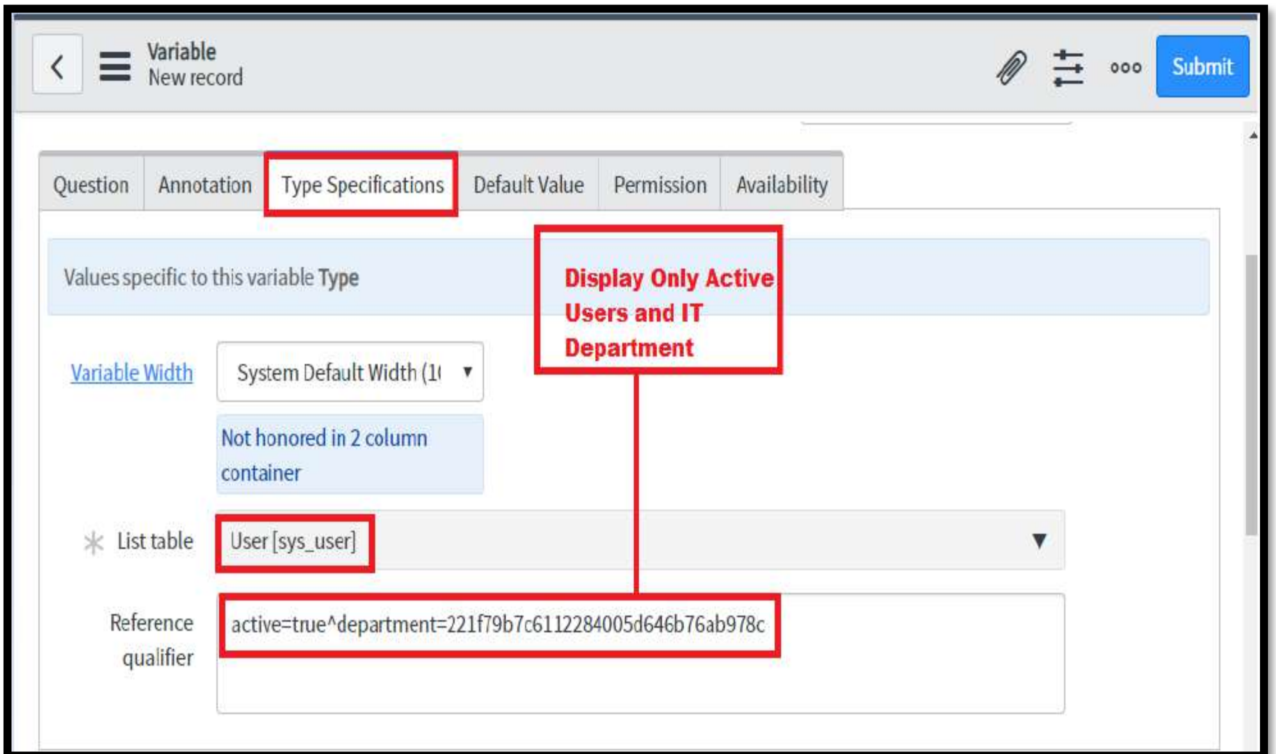


5. Fill **List Collector** variable form

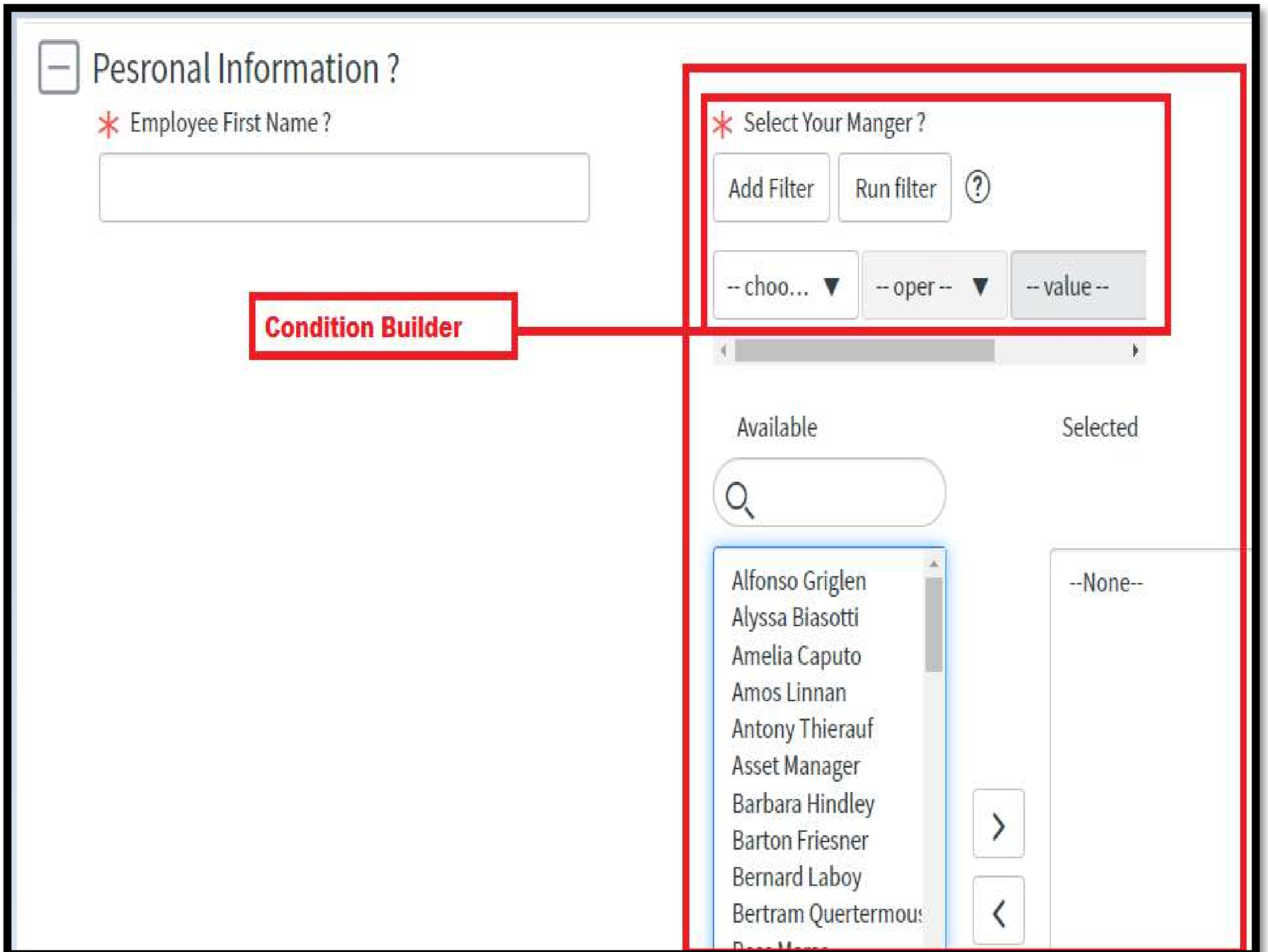


6. **Variable Type:** List Collector

7. **CatLog Item:** Auto Filled
8. **Mandatory:** Checked **True**
9. **Order:** 300
10. **Name:** Auto Filled
11. **Question:** Select Your Manger
12. **Toll Tip:** Select your Reporting Manager



13. Open Type **Specification Tab**
14. **List Table: User[sys_user]**
15. **reference qualifies:** active=true^department=221f79b7c6112284005d646b76ab978c
16. Click on **Submit**



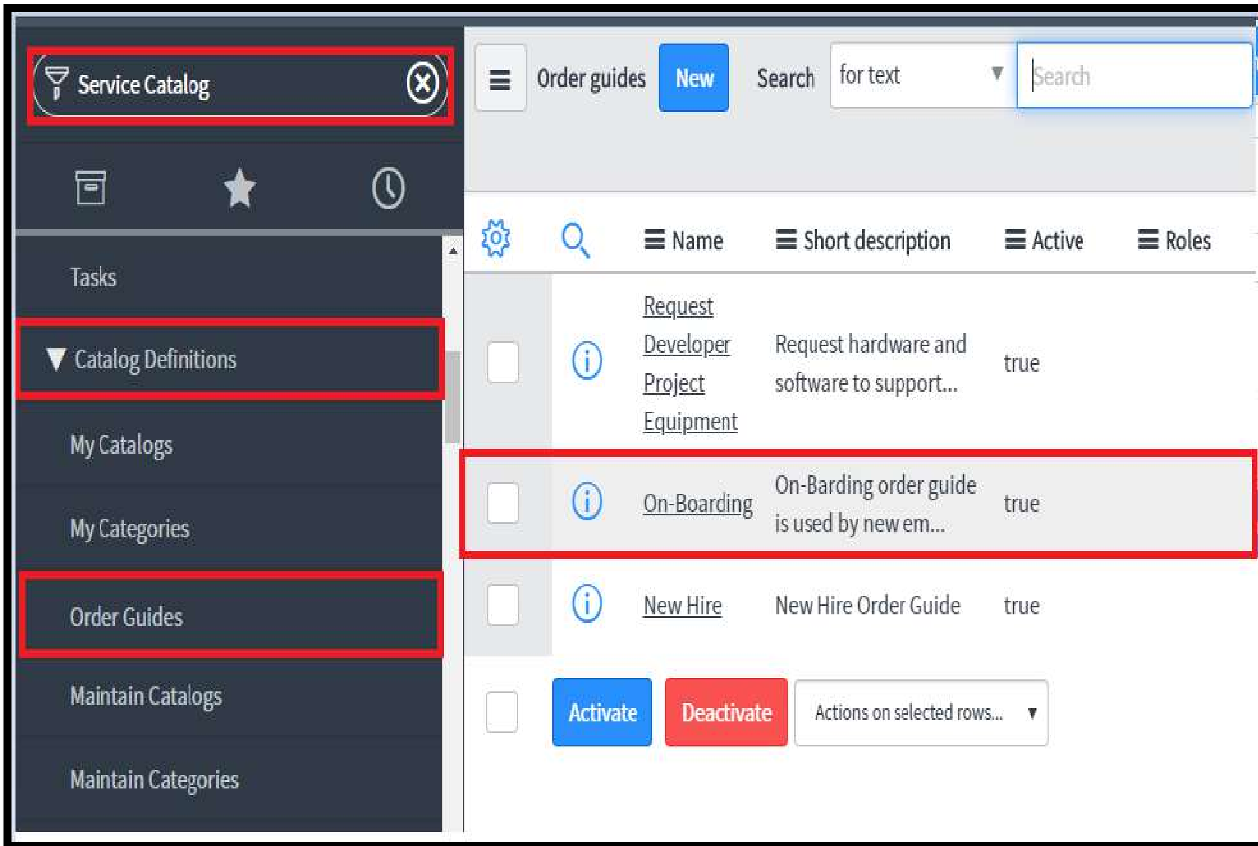
17. Click on **Try It** to preview the **Order Guide**

Working with Multi Line Text Variable

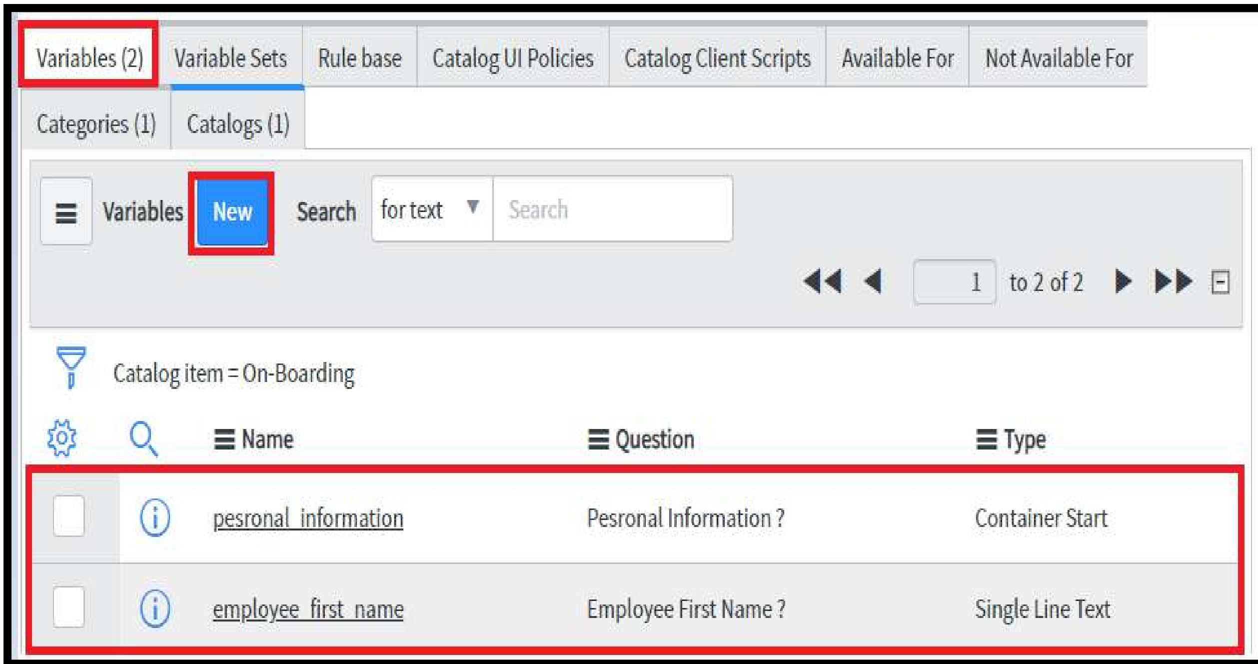
This variable is used get multi line of text from customer

Procedure

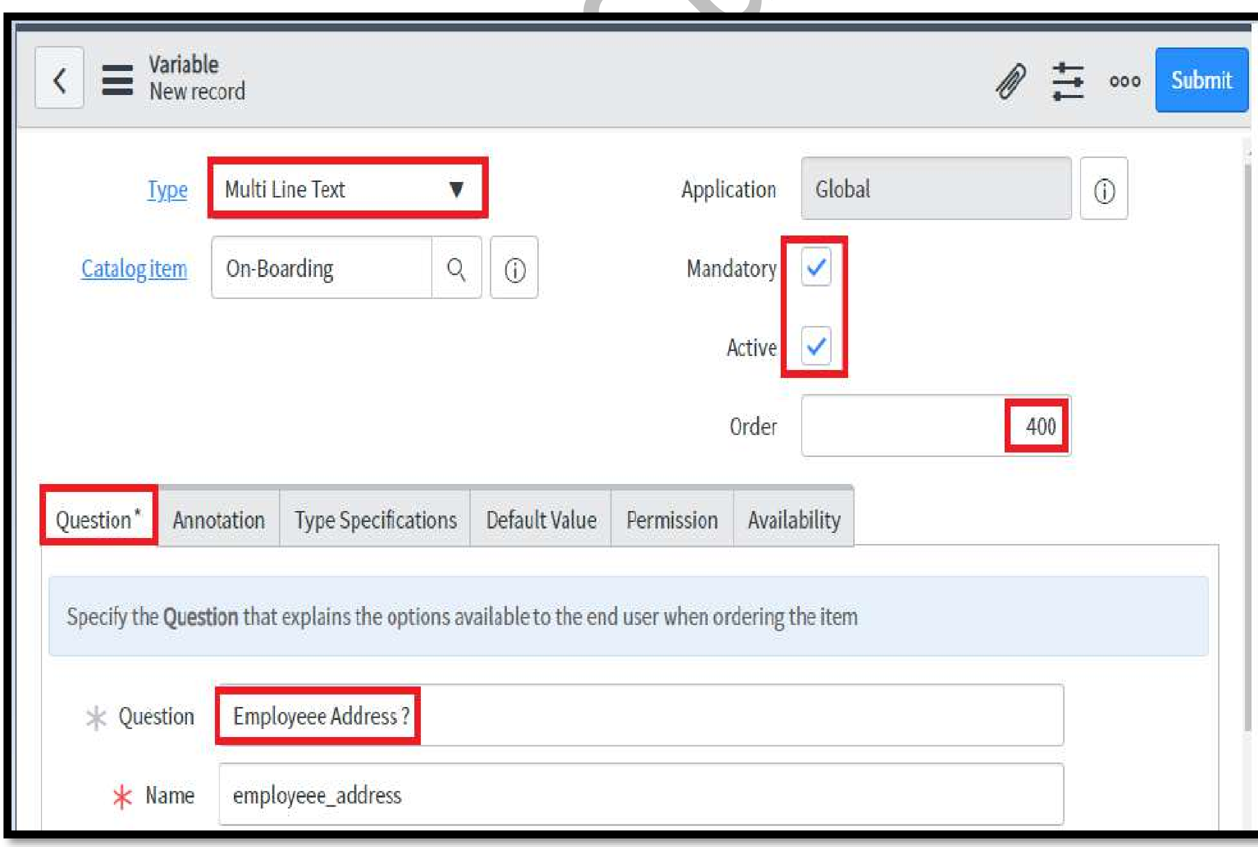
1. Navigate to **Service Catalog** > **Catalog Definition** > **Order Guides**
2. Open **On-Boarding** order guide Record



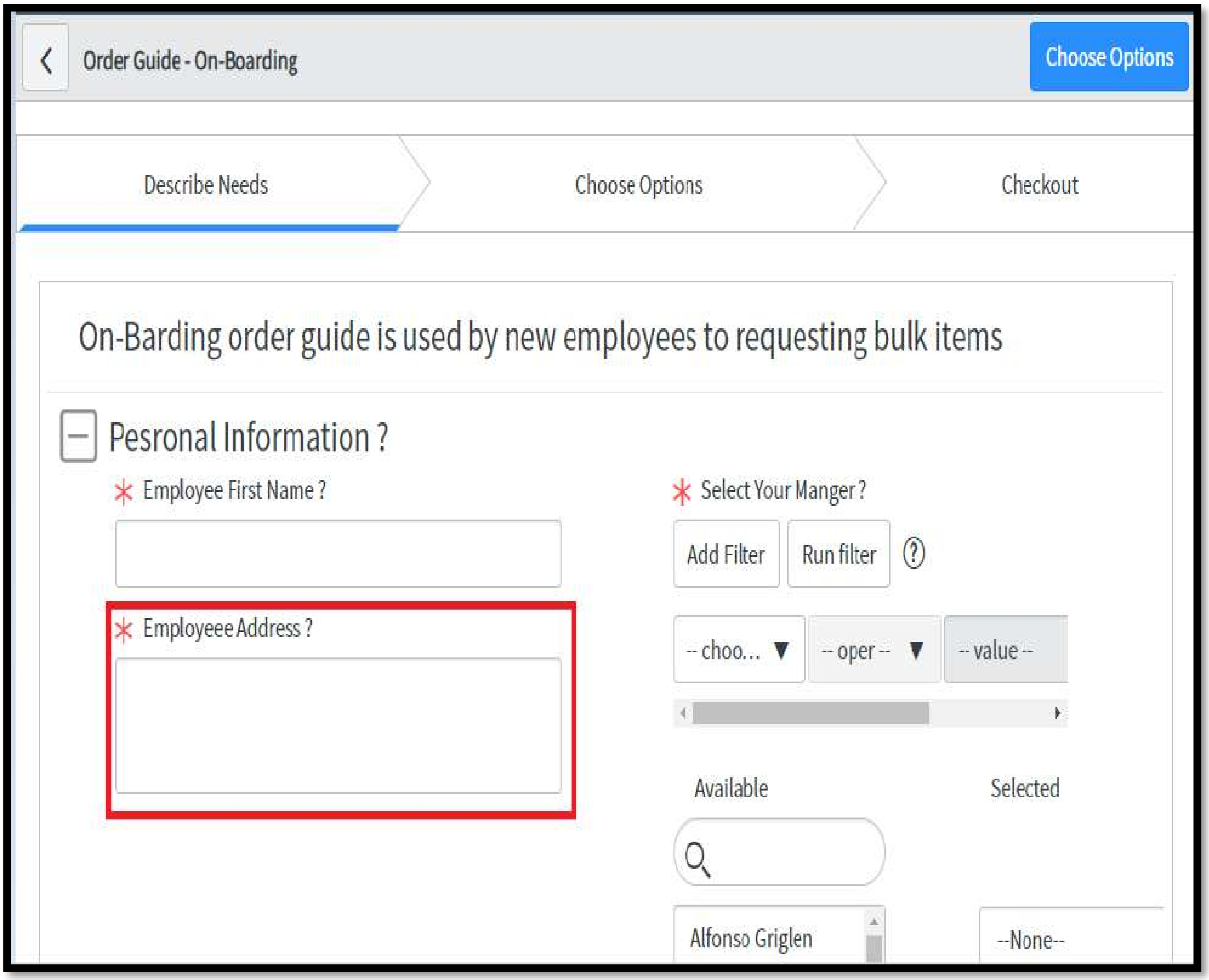
3. Open **Variable Tab** from Related List
4. Click on **New**



5. Fill **Multi Line Text** variable form



6. **Variable Type: Multi Line Text**
7. **Mandatory:** Checked **True**
8. **Order:** 400
9. **Question:** Employee Address
10. **Name:** Auto Filled
11. Click on **Submit**



1. Click on **Try It** to preview the **Order Guide**

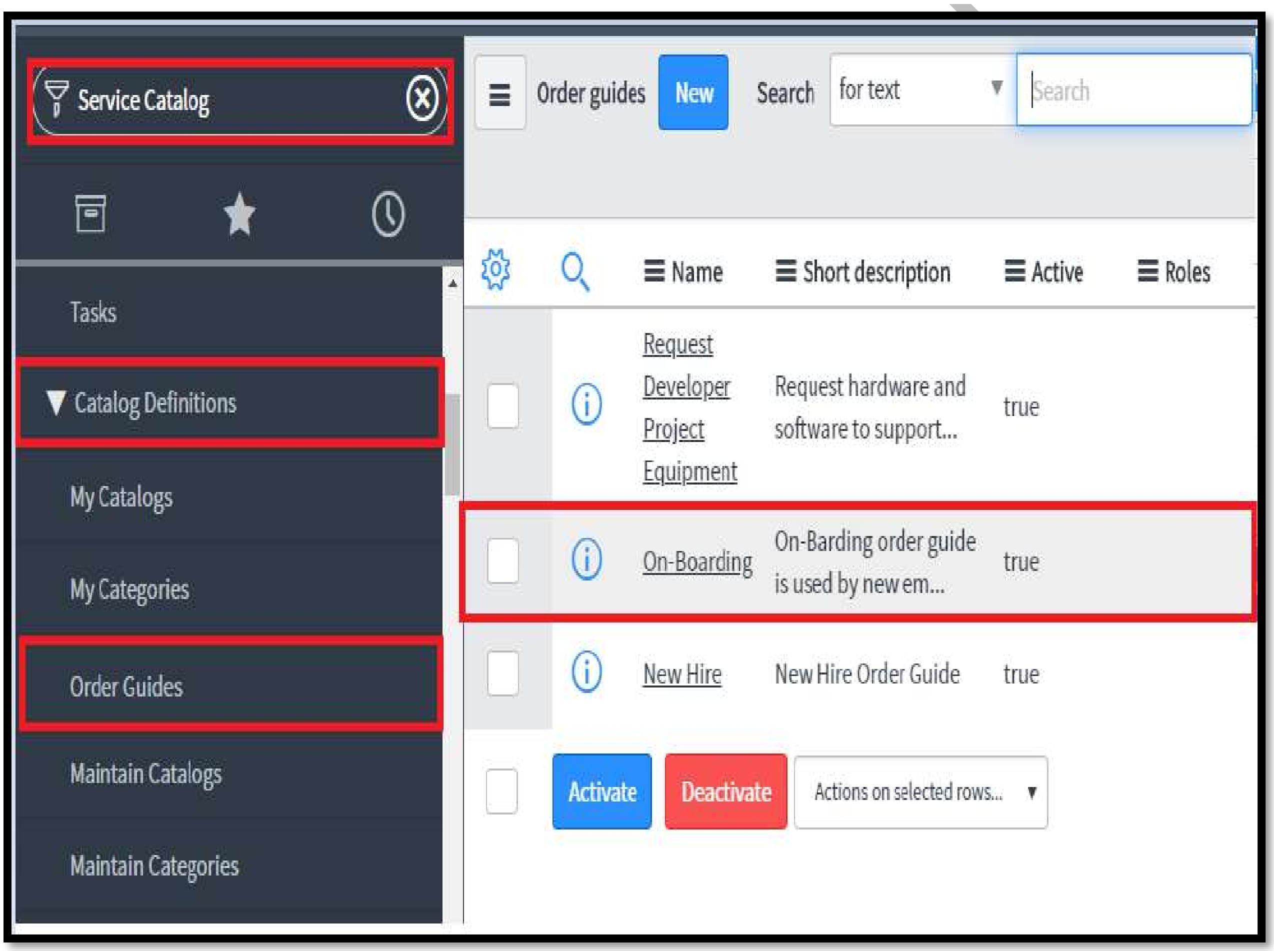
Working with Label Variable

Label Variable can display a static information on the form. Labels are form elements. This variable not collect any data

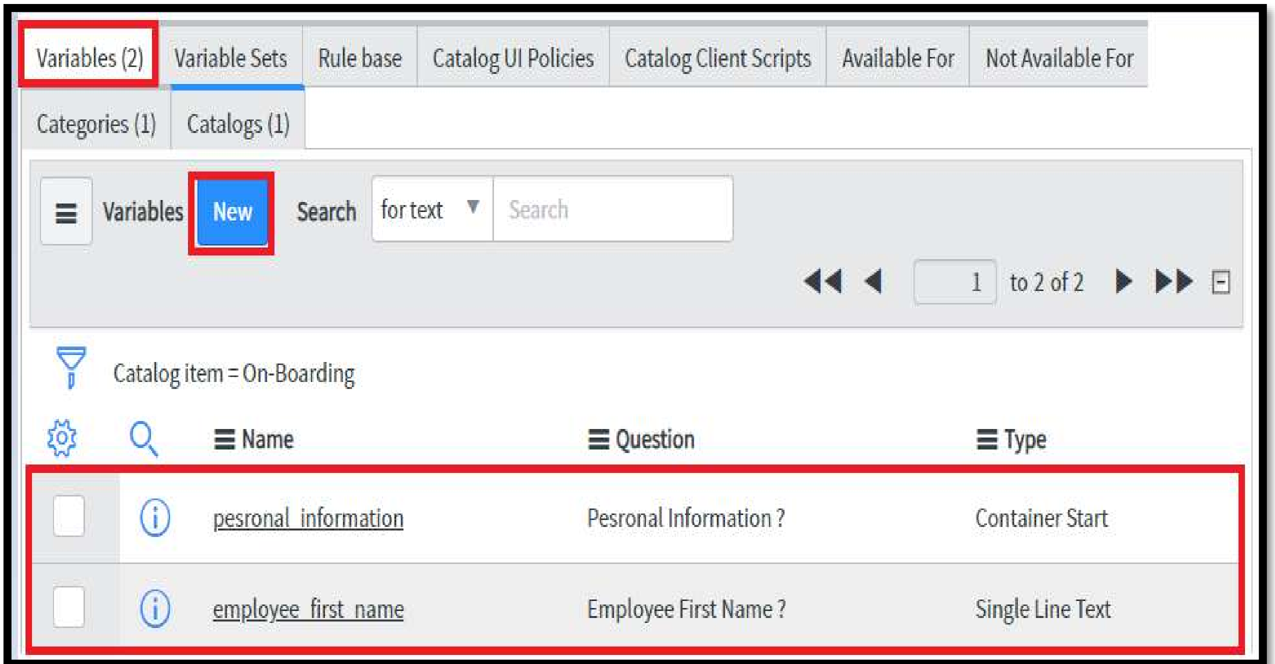
Note: We can use labels variable to arrange questions into different sections. Use labels to create a multi-option question (for example, using check box Variables).

Procedure

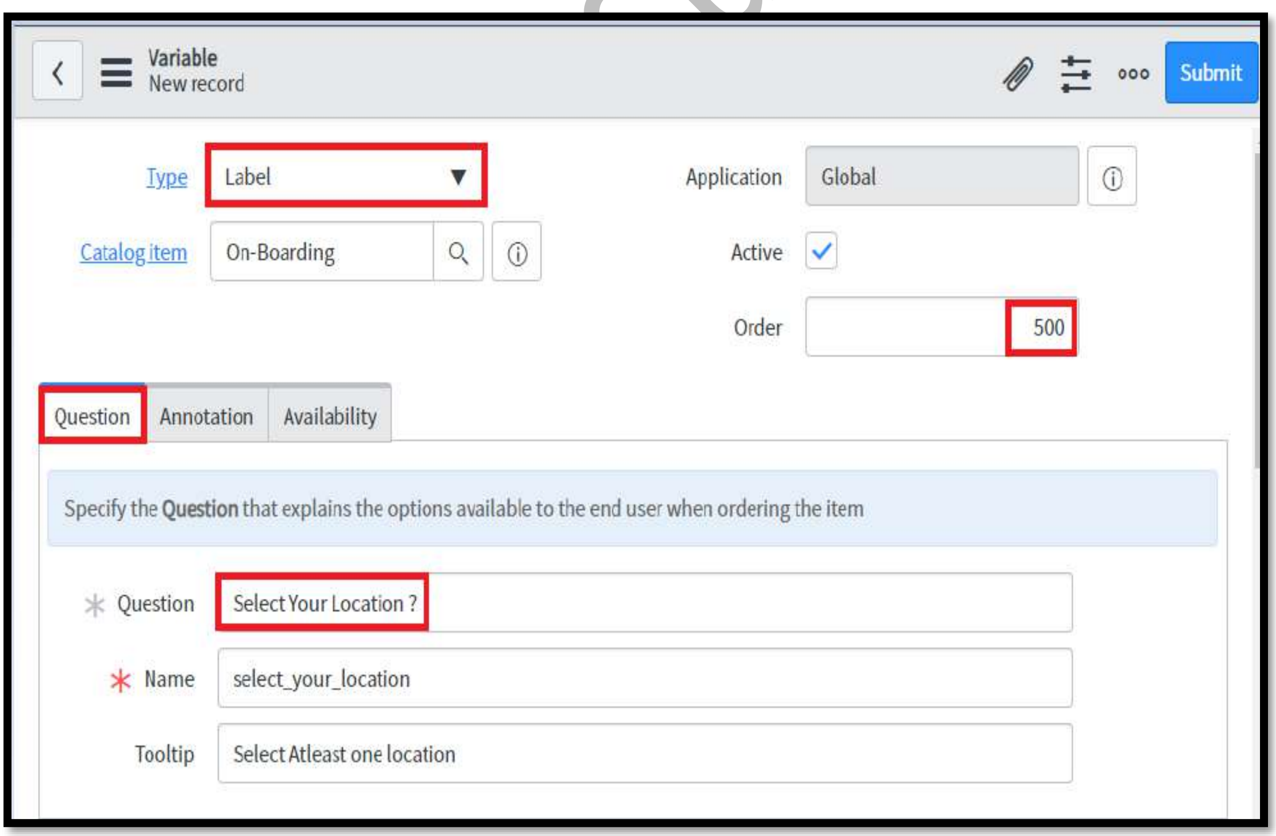
1. Navigate to **Service Catalog** > **Catalog Definition** > **Order Guides**
2. Open **On-Boarding** order guide Record



3. Open **Variable Tab** from Related List
4. Click on **New**

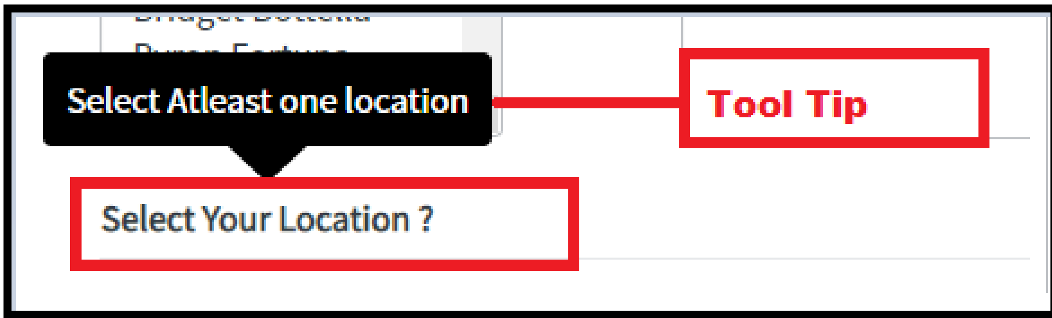


5. Fill **Lebel** variable form



6. **Variable Type:**

7. **Order:** 500
8. **Question:** Select Your Location
9. **Name:** Auto Filled
10. **Tooltip:** Select At least one address
11. Click on **Submit**



12. Click on **Try It** to preview the **Order Guide**

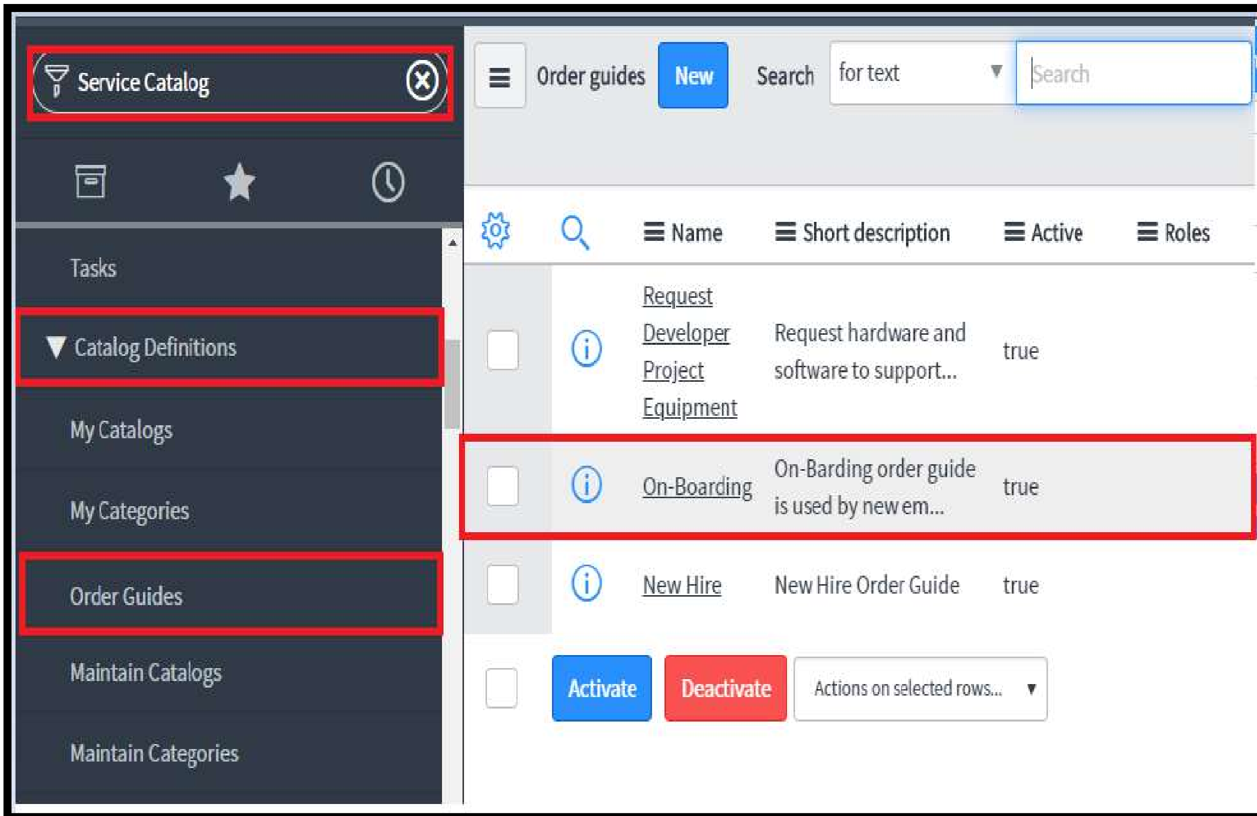
Working with Check Box Variable

Check Box variable is used to **enable/disable** options by **checked/unchecked**. List of check boxes in order to display under a **label** to create a multi-option

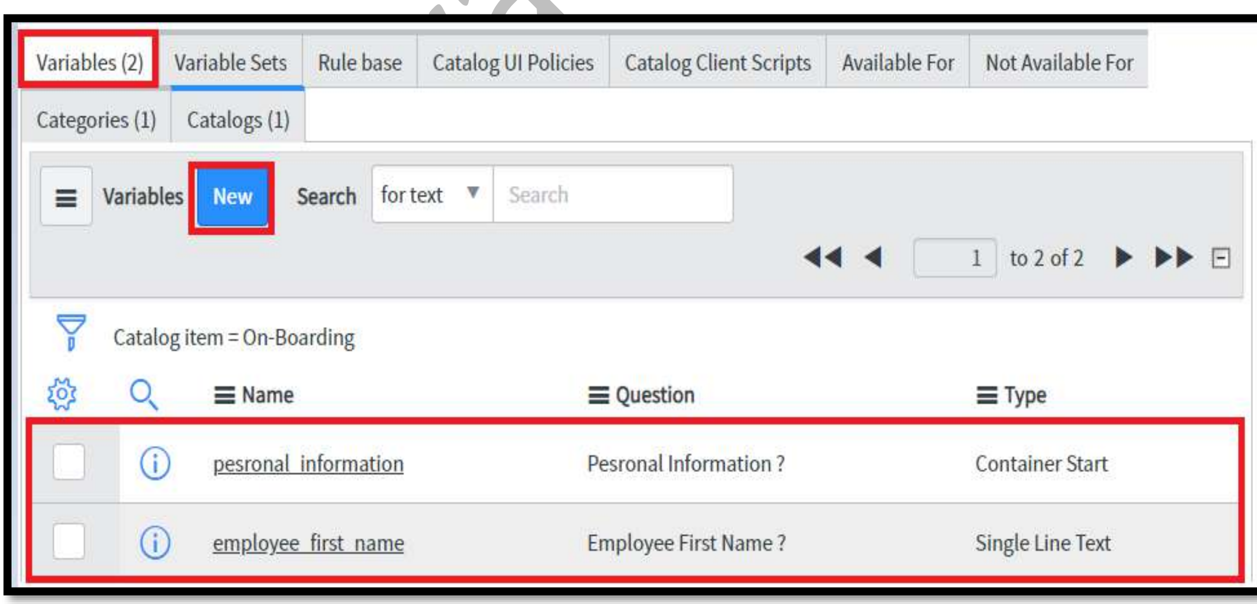
Number check box variables are automatically grouped under an **Options label**. If you want to define and display a label of our own, then group check boxes under a **label** variable. The default Options label is replaced with the label variable.

Procedure

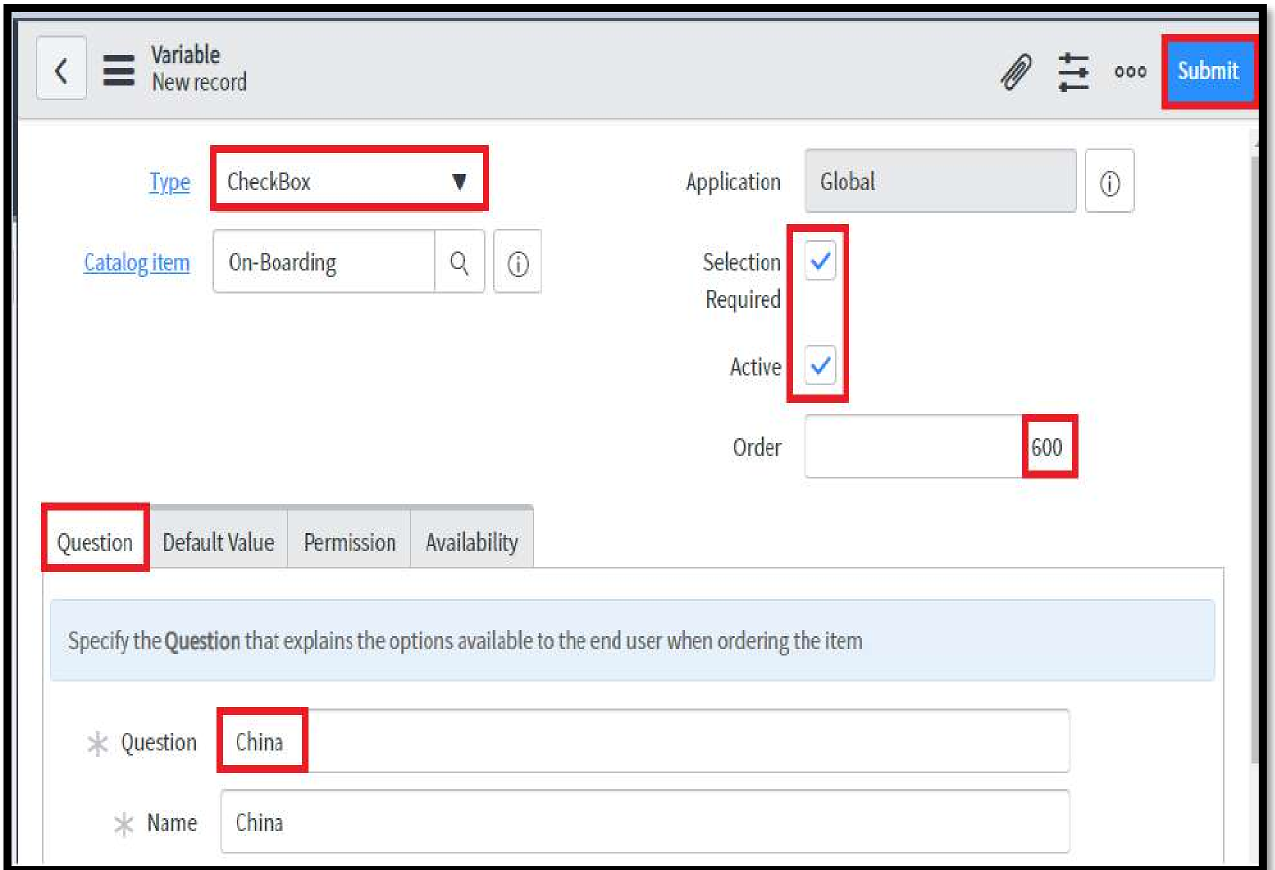
1. Navigate to **Service Catalog** > **Catalog Definition** > **Order Guides**
2. Open **On-Boarding** order guide Record



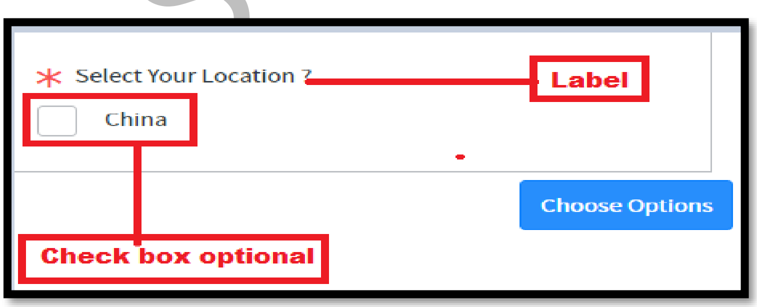
3. Open **Variable Tab** from Related List
4. Click on **New**



5. Fill **Check Box** variable form



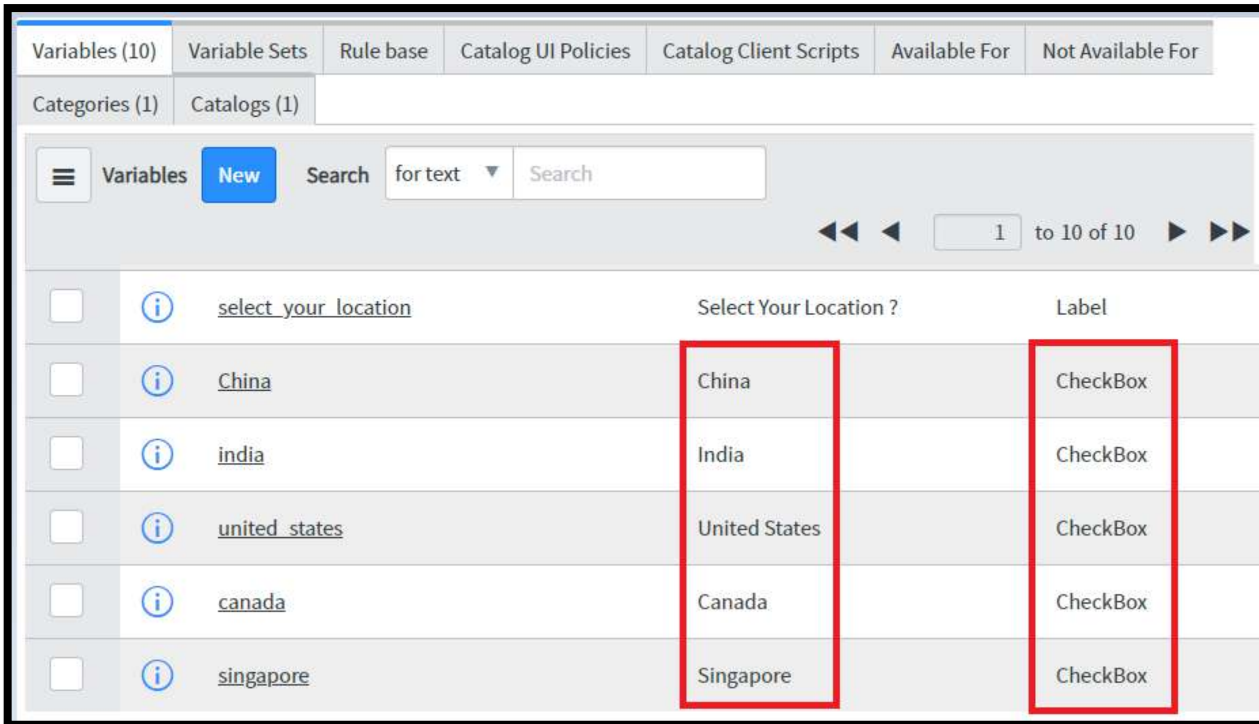
6. Variable Type: Check Box
7. **Catalog Item**: On-Board **(Auto Filled)**
8. **Selection Required:** Checked True
9. **Order:** 600
10. **Active:** Checked True
11. **Question:** China
12. **Name:** Auto Filled
13. Click on **Submit**



14. Click on **Try It** to preview the **Order Guide**

Similarly Create 4 more Check Box variables

15. We have to create more checkbox variables in **Order Guide** to provide more options for customers. **(Refer below screen shot to create more**



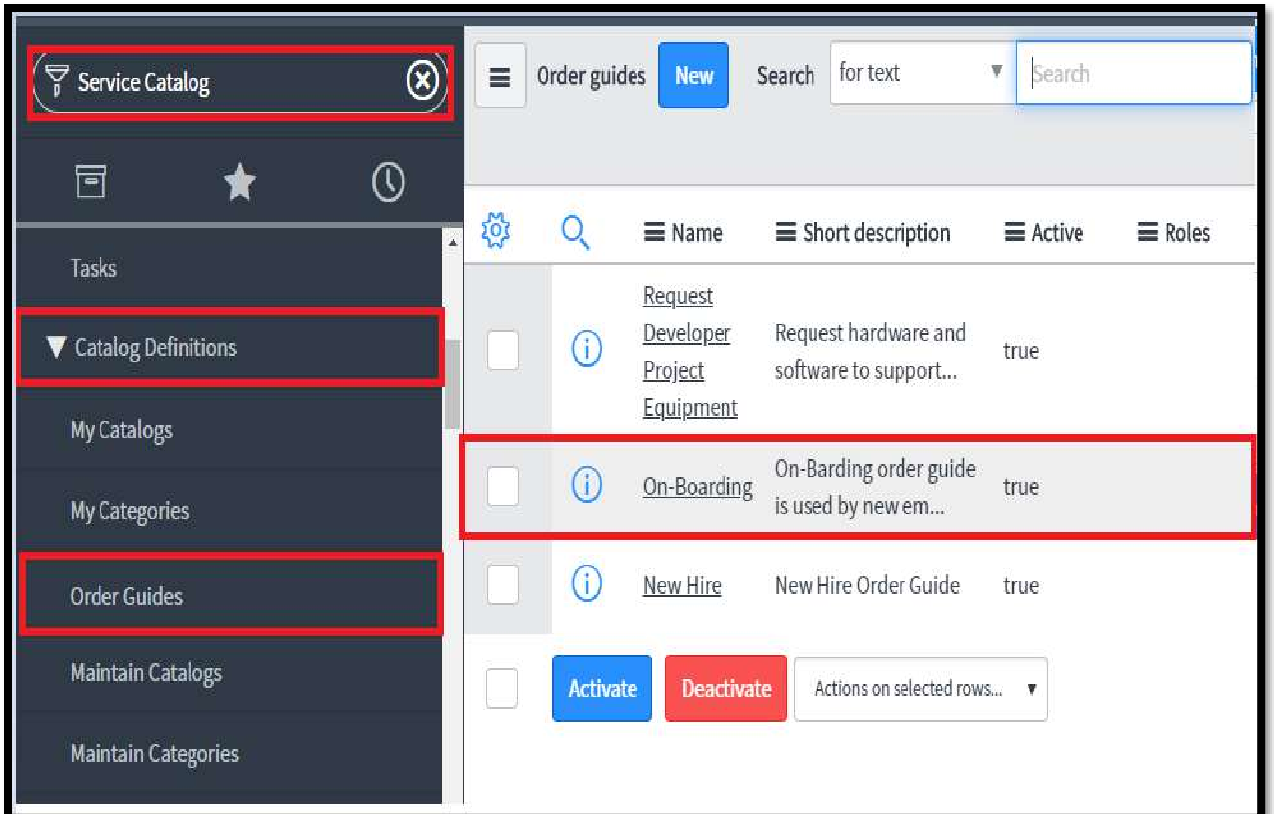
16. Click on **Try It** to preview the **Order Guide**

Working with Select Box Variable

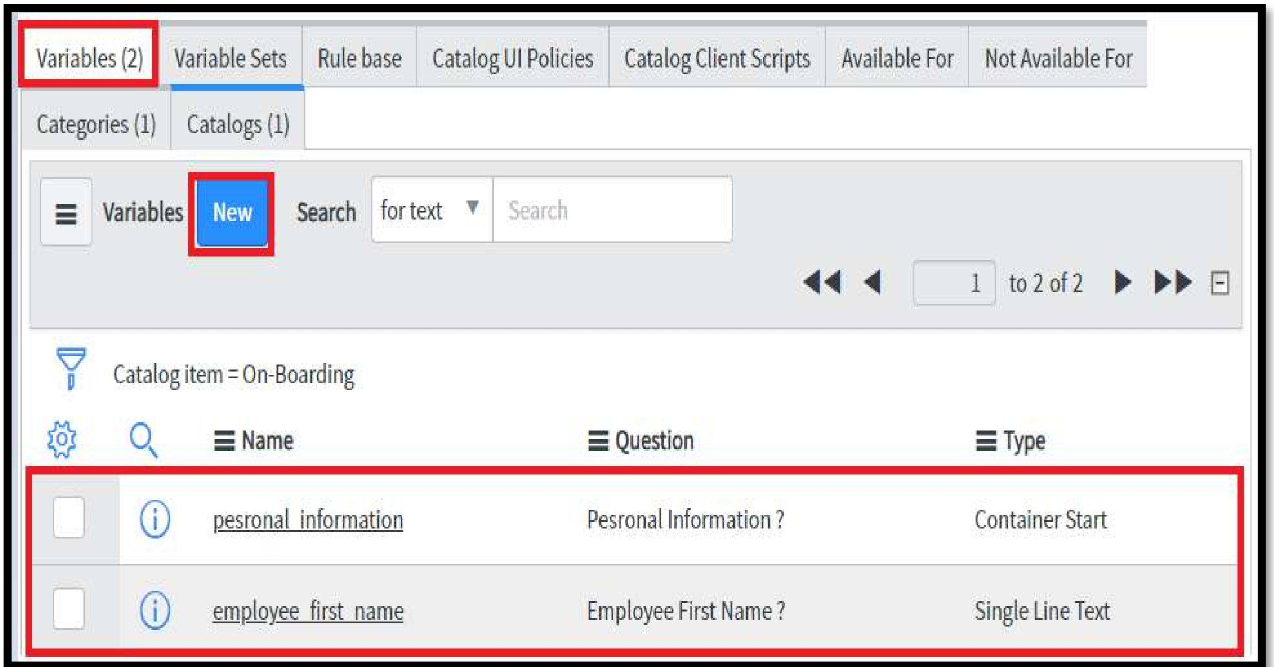
The select box variable is used to a list from predefined question choices.

Procedure

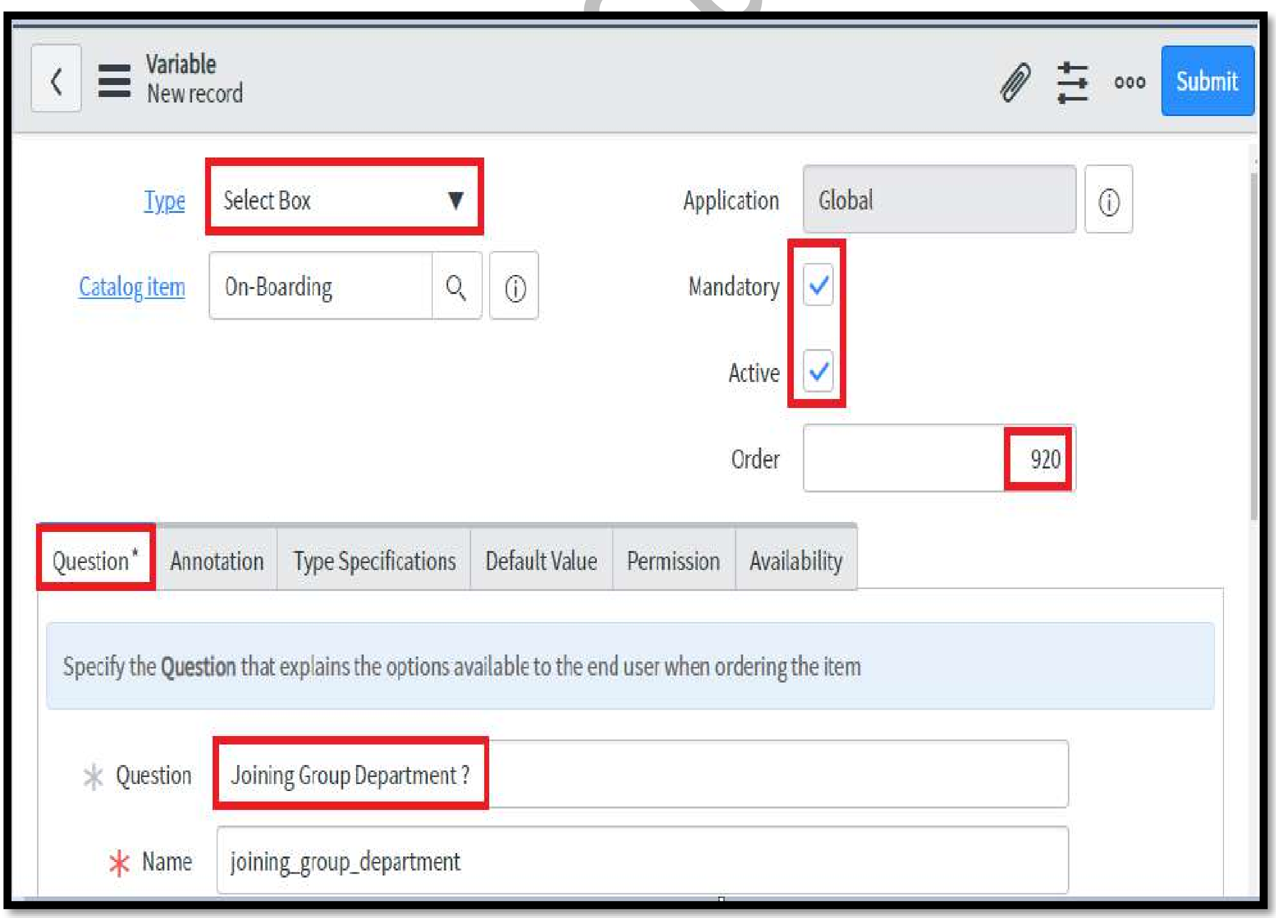
1. Navigate to **Service Catalog** > **Catalog Definition** > **Order Guides**
2. Open **On-Boarding** order guide Record



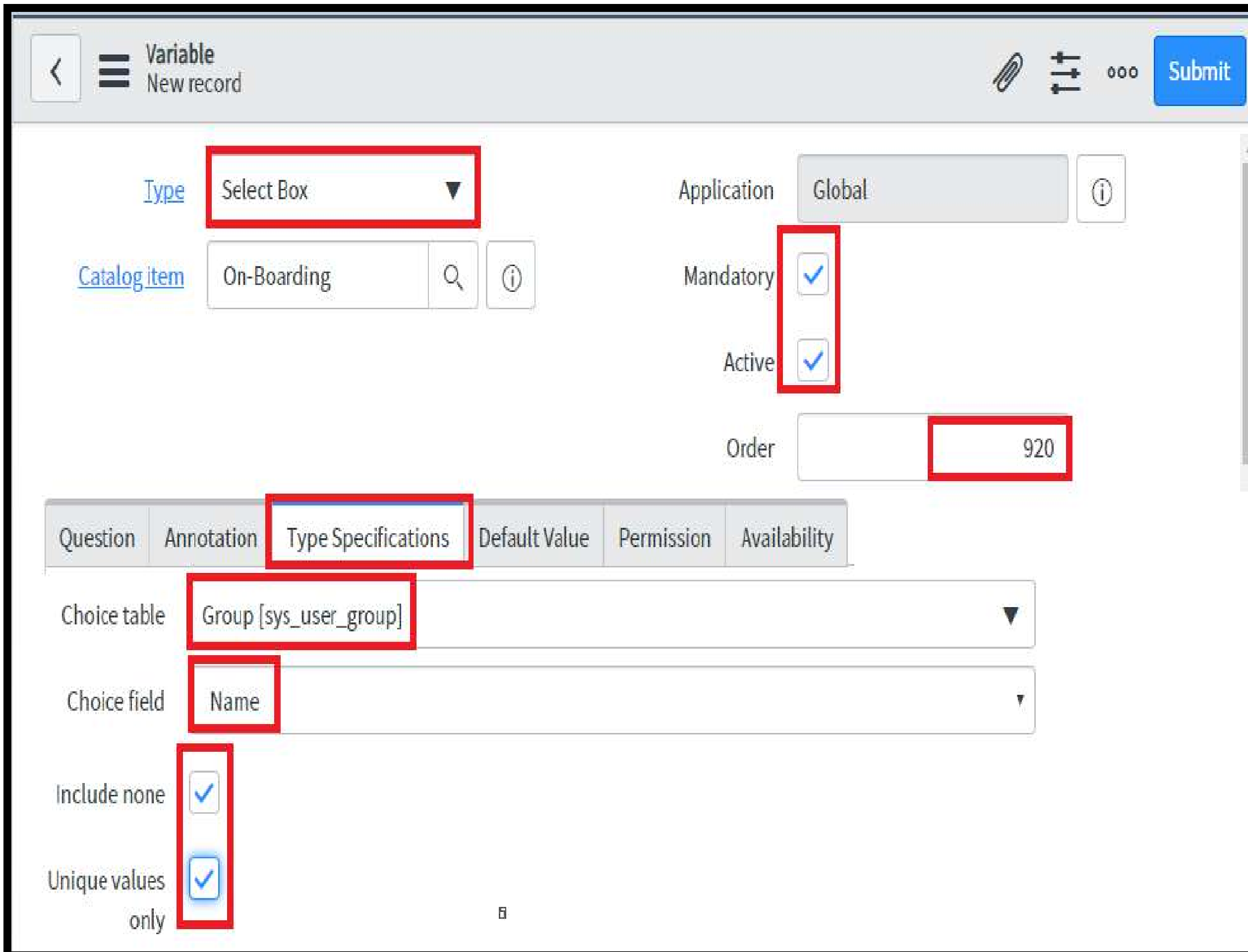
3. Open **Variable Tab** from Related List
4. Click on **New**



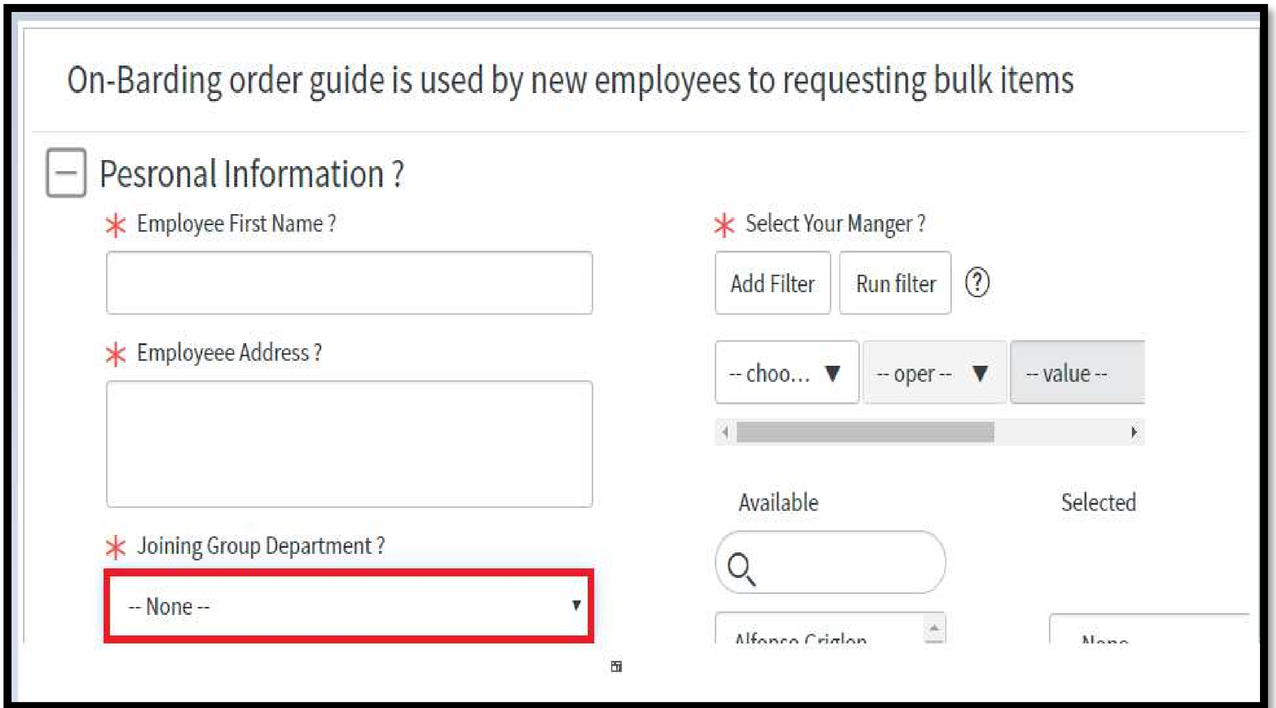
5. Fill **Select Box** variable form



6. **Variable Type:** Select Box
7. **Mandatory:** Checked True
8. **Order:** 920
9. **Question:** Joining Group Department?
10. **Name:** Auto Filled



11. Open **Type Specification** Tab
12. Choice Table: Group**[sys_user_group]**
13. **Choice field:** Name
14. **Include none:** Checked **True**
15. **Unique values only:** Checked **True**
16. Click on **Submit**



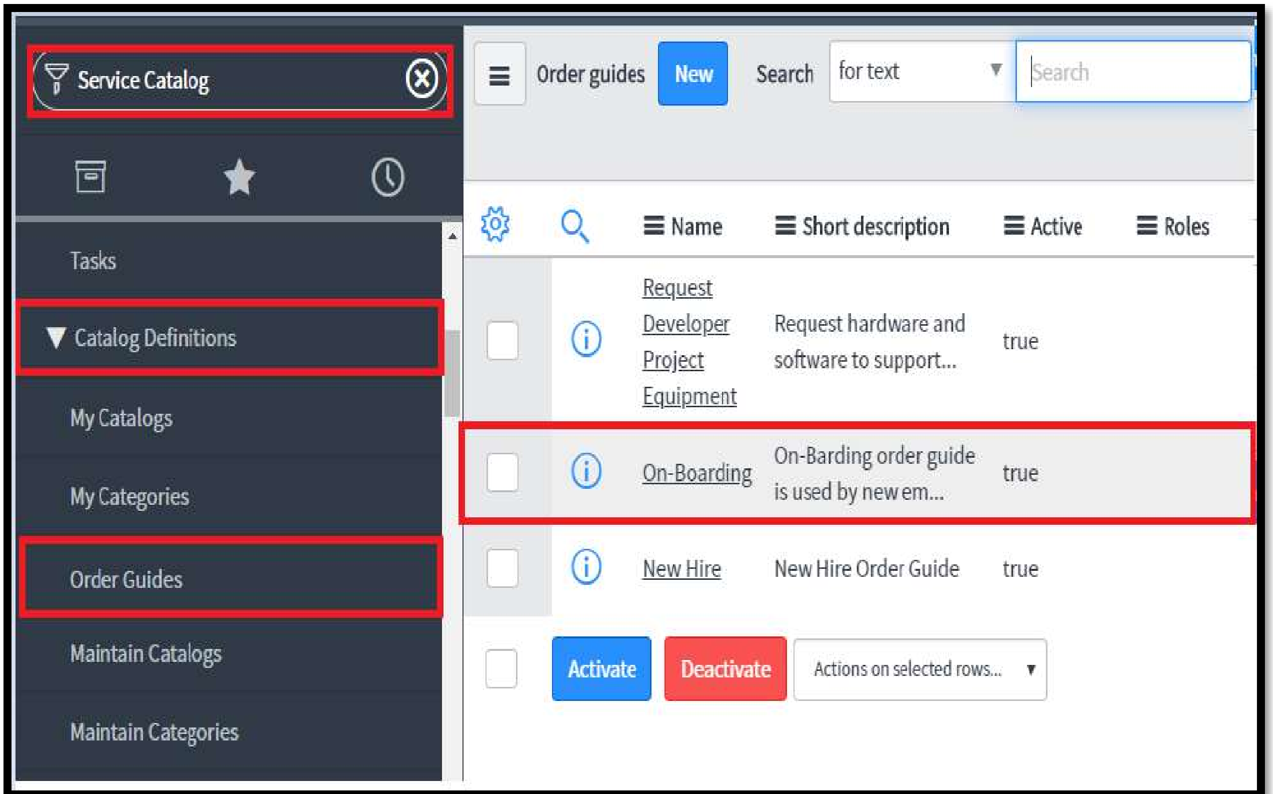
17. Click on **Try It** to preview the **Order Guide**

Working with Yes/No Variable

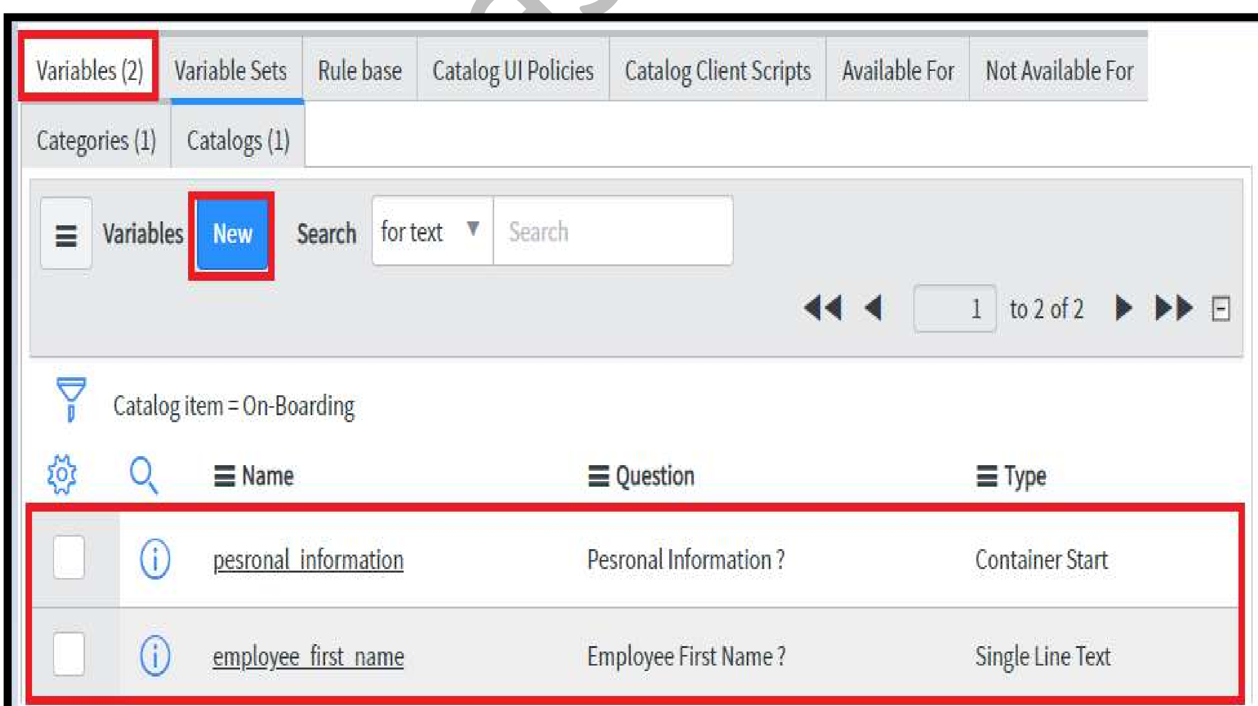
The **Yes or No** variable provide choice list with **Yes** and **No** as options. Including None also

Procedure

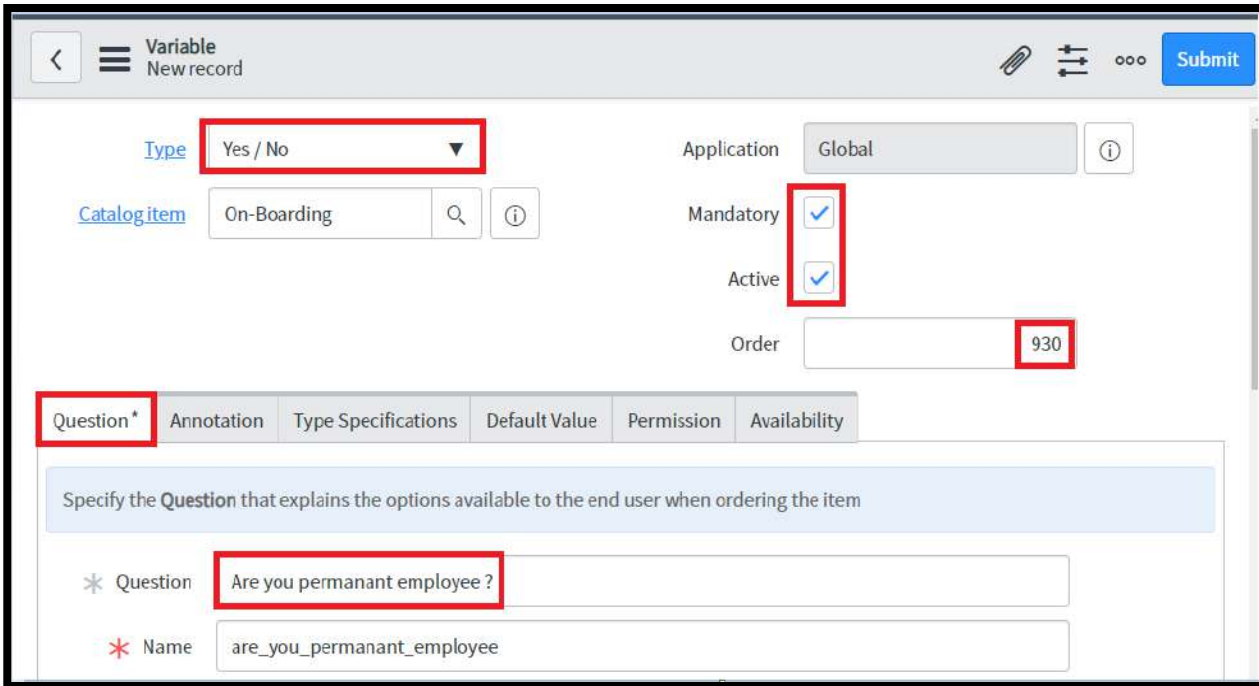
1. Navigate to **Service Catalog** > **Catalog Definition** > **Order Guides**
2. Open **On-Boarding** order guide Record



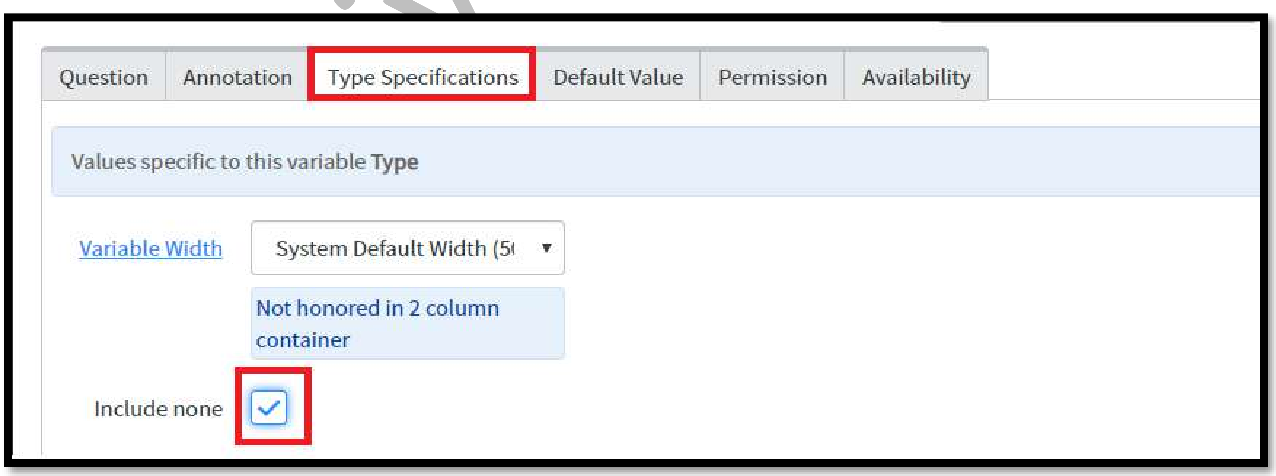
3. Open **Variable Tab** from Related List
4. Click on **New**



5. Fill **Yes/No** variable form



6. **Variable Type:** Yes/No
7. **Mandatory:** Checked True
8. **Order:** 930
9. **Question:** Are you permanent employee?
10. **Name:** Auto Filled



11. Open **Type Specification** Tab
12. Include none: Checked **True**
13. Click on **Submit**

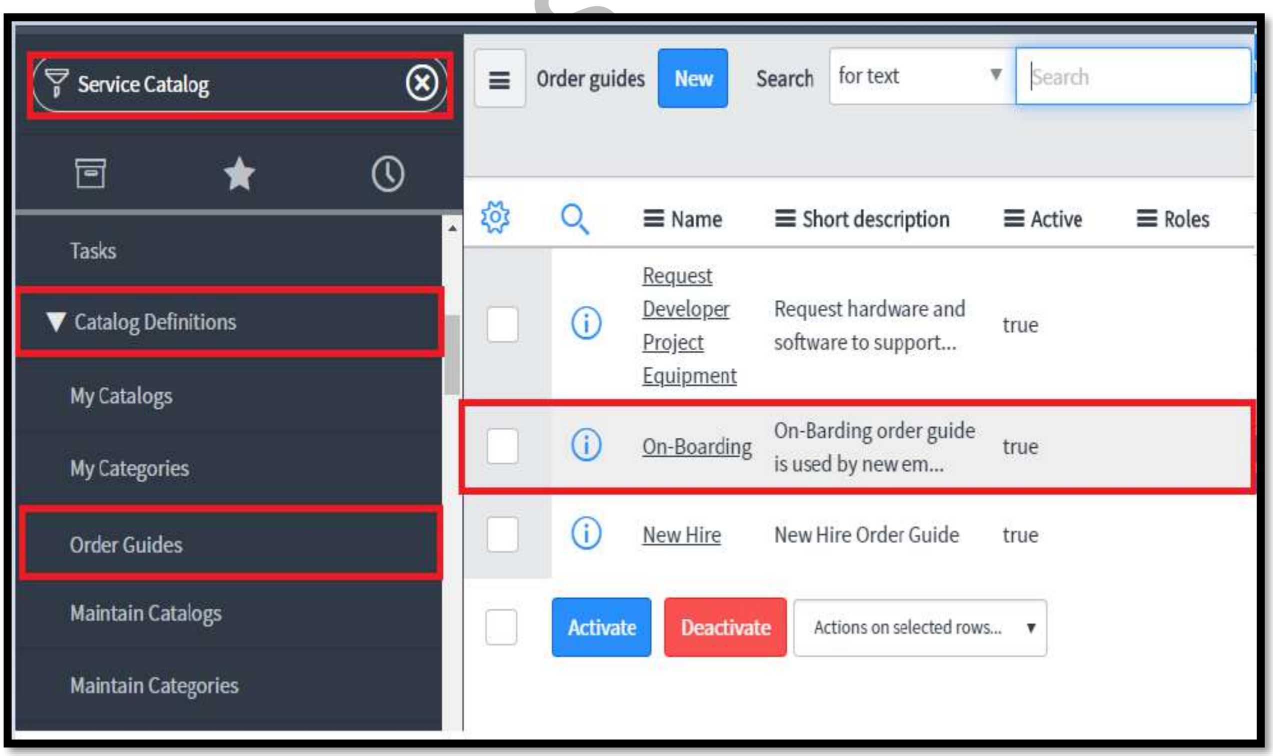
14. Click on **Try It** to preview the **Order Guide**

Working with Yes/No Variable

The **Yes or No** variable provide choice list with **Yes** and **No** as options. Including None also

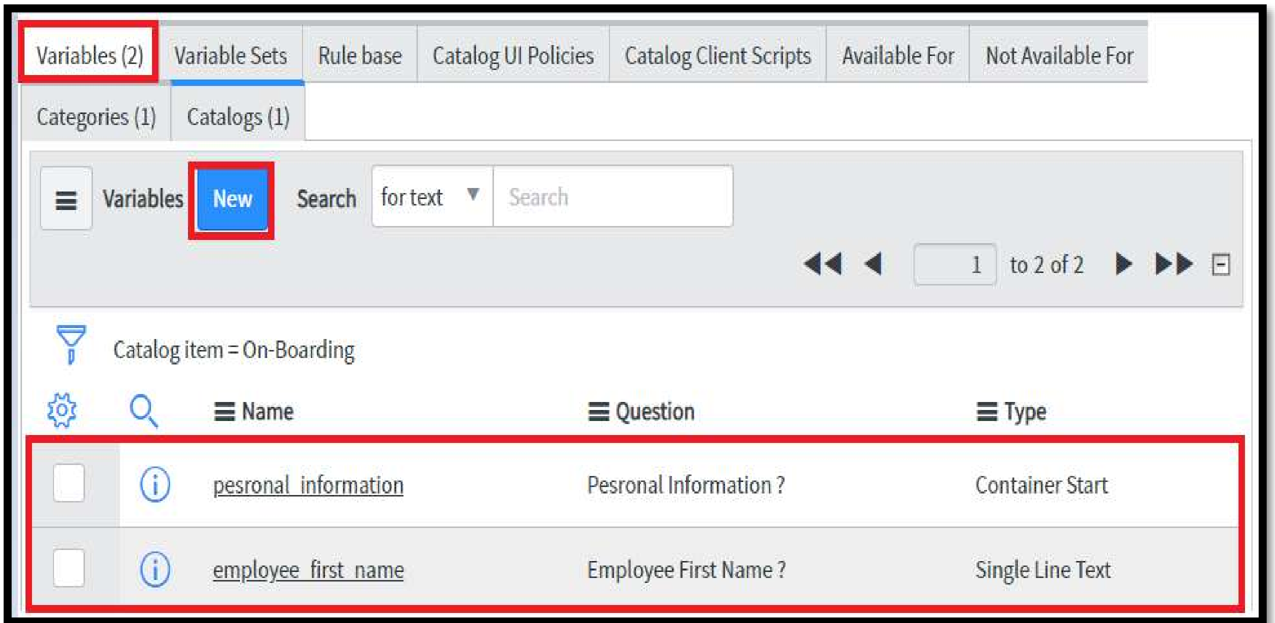
Procedure

1. Navigate to **Service Catalog** > **Catalog Definition** > **Order Guides**
2. Open **On-Boarding** order guide Record

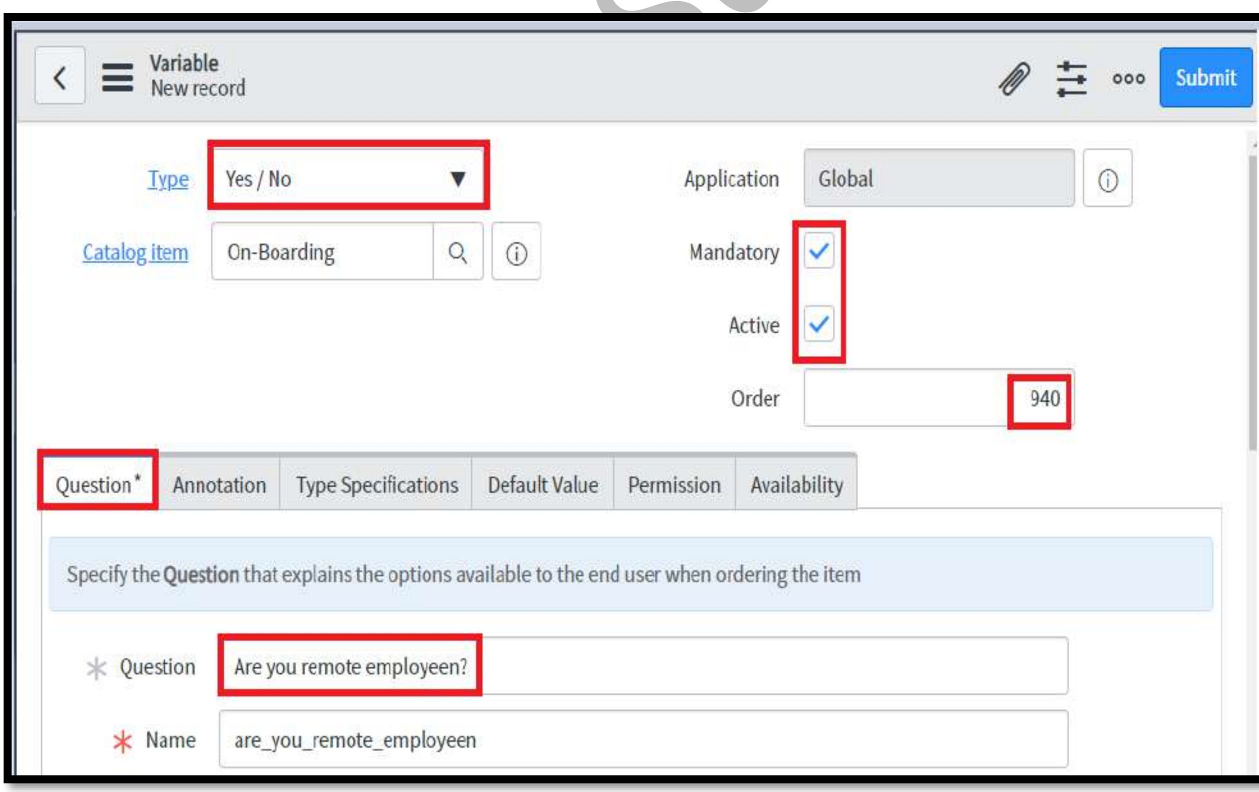


3. Open **Variable Tab** from Related List

4. Click on **New**

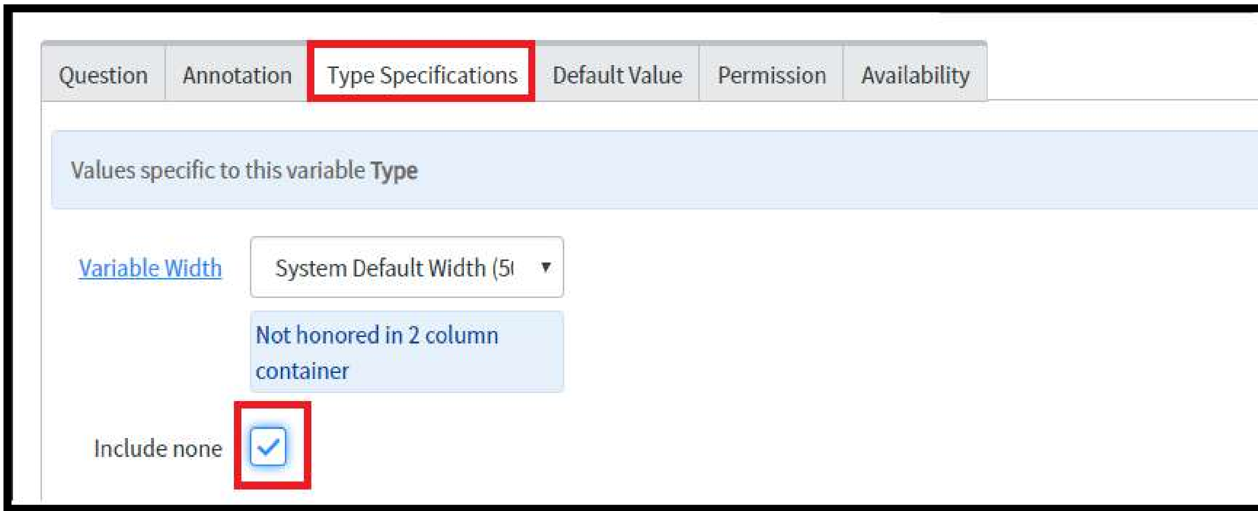


5. Fill **Yes/No** variable form

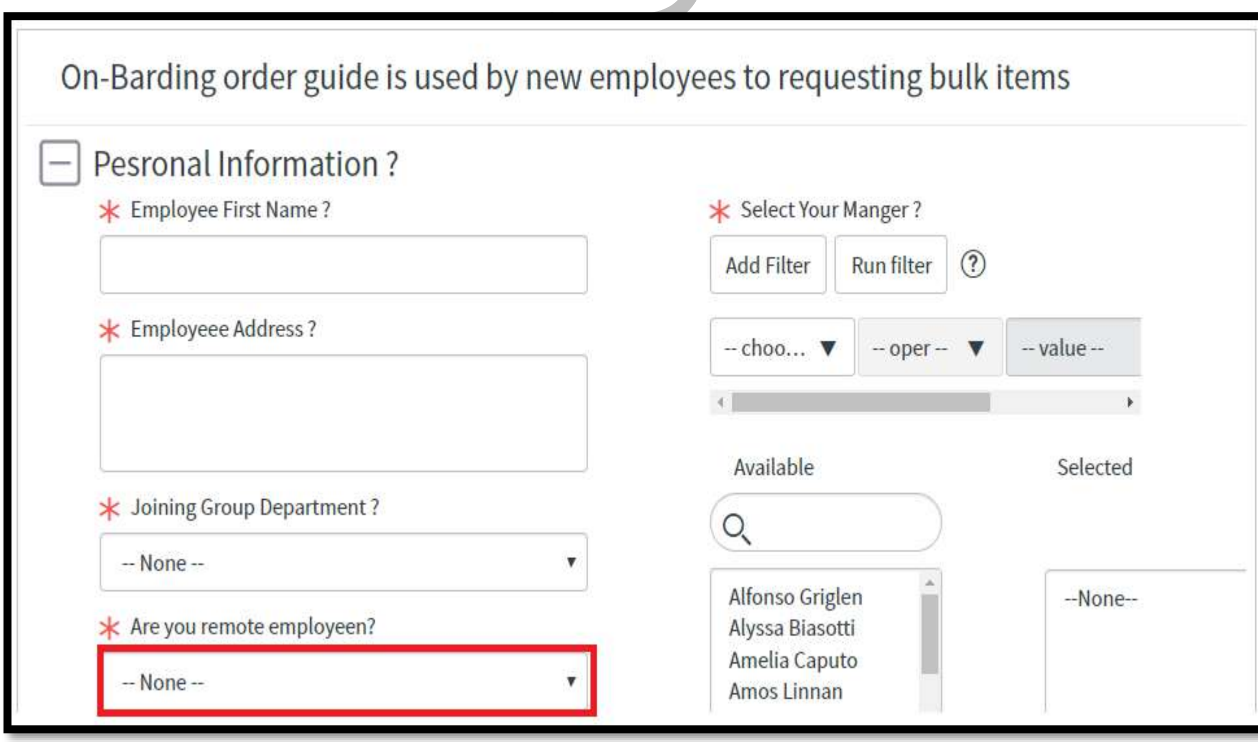


6. **Variable Type:** Yes/No
7. **Mandatory:** Checked True

8. **Order:** 940
9. **Question:** Are you remote employee?
10. **Name:** Auto Filled



11. Open **Type Specification** Tab
12. Include none: Checked **True**
13. Click on **Submit**



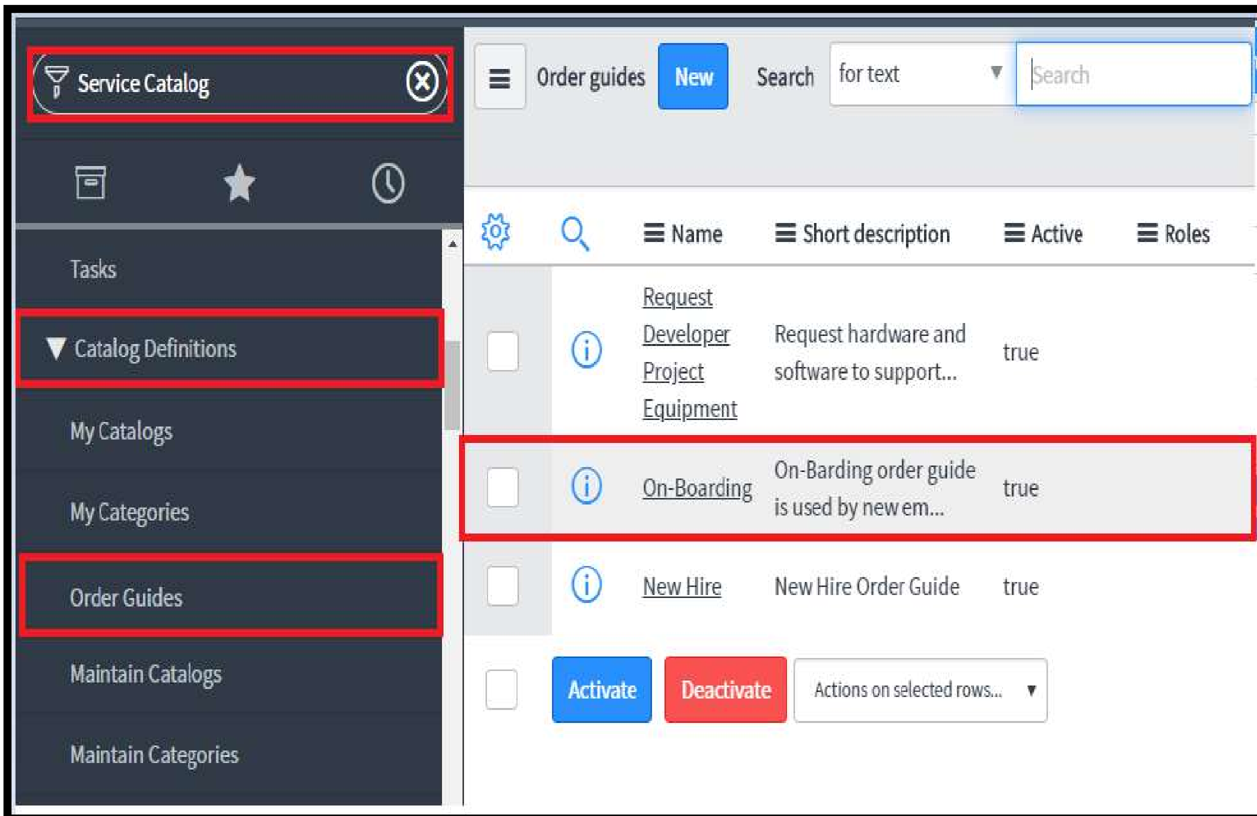
14. Click on **Try It** to preview the **Order Guide**

Working with Container End Variable

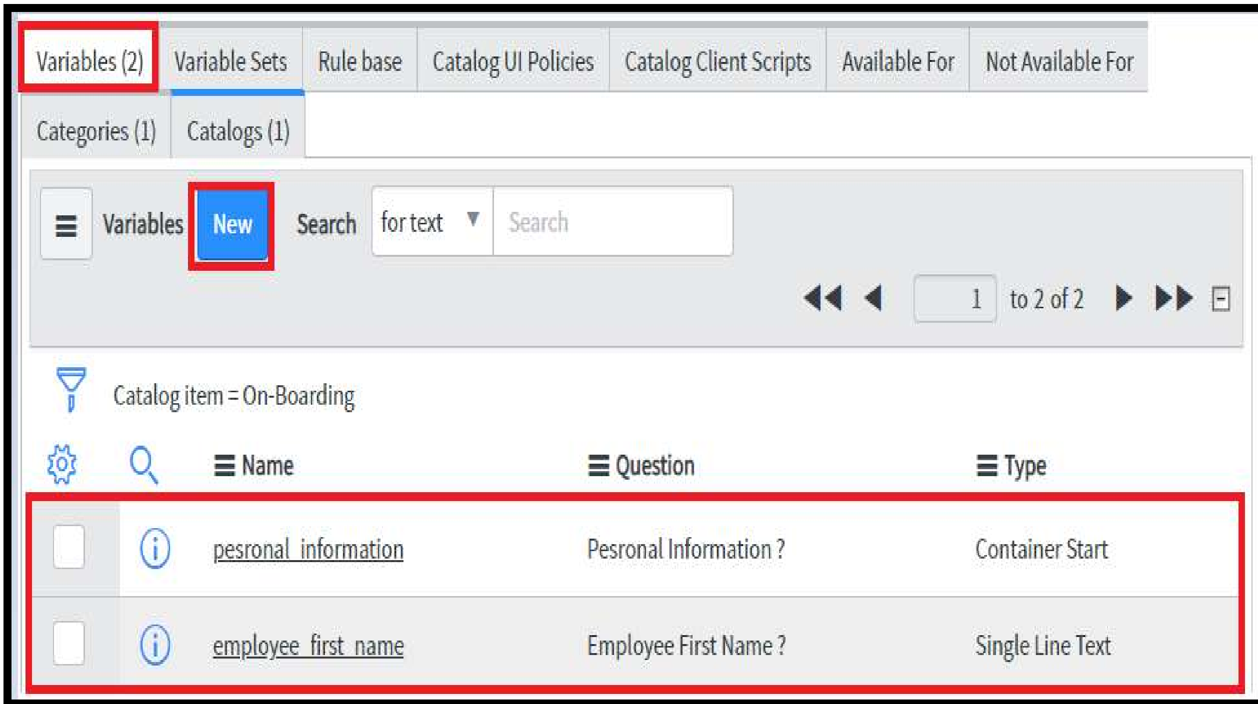
The Container **End** variable should be used along with **Container Start** variable to close a container

Procedure

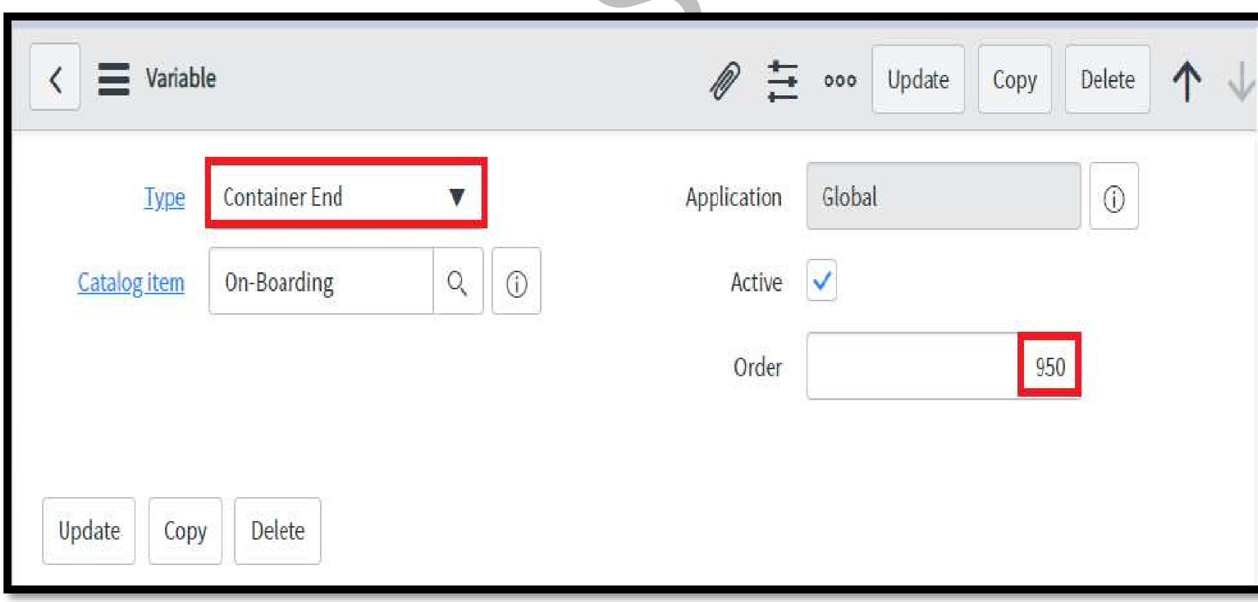
1. Navigate to **Service Catalog** > **Catalog Definition** > **Order Guides**
2. Open **On-Boarding** order guide Record



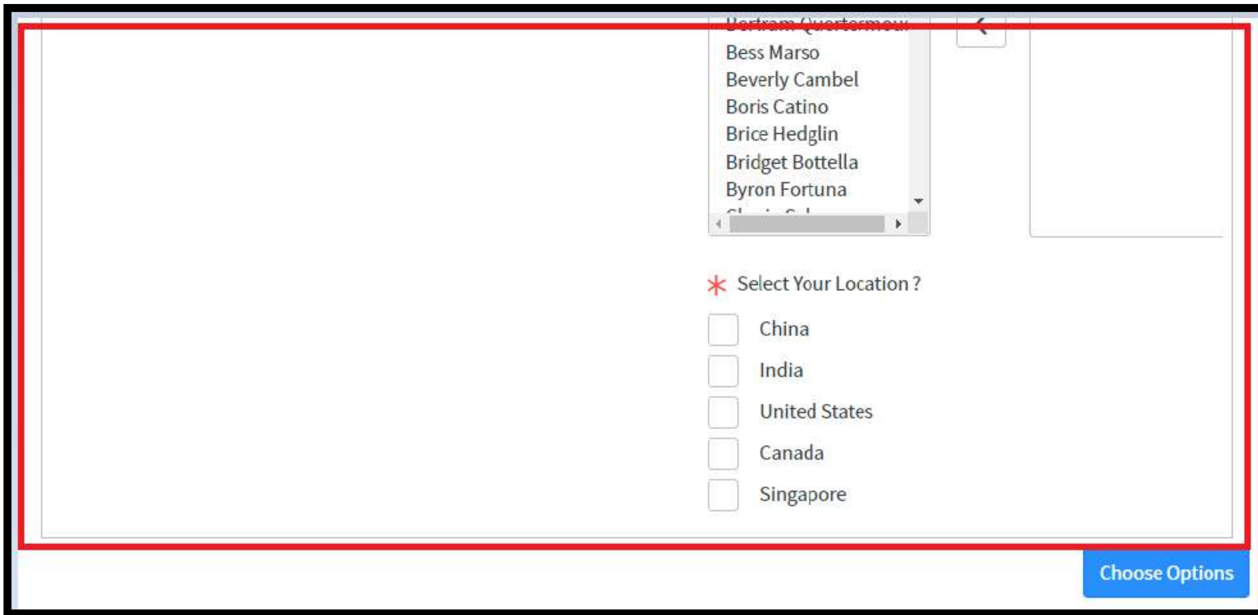
3. Open **Variable Tab** from Related List
4. Click on **New**



5. Fill **Container End** variable form



6. **Variable Type:** Container End
7. **Order:** 950
8. Click on **Submit**



Exercise: 8 Configure Rule Base in Order Guide

We can add some rules for catalog item to an order guide based on our client requirement

Order Guide Rules can be defining the conditions that should be met for a specific item to be included in an order.

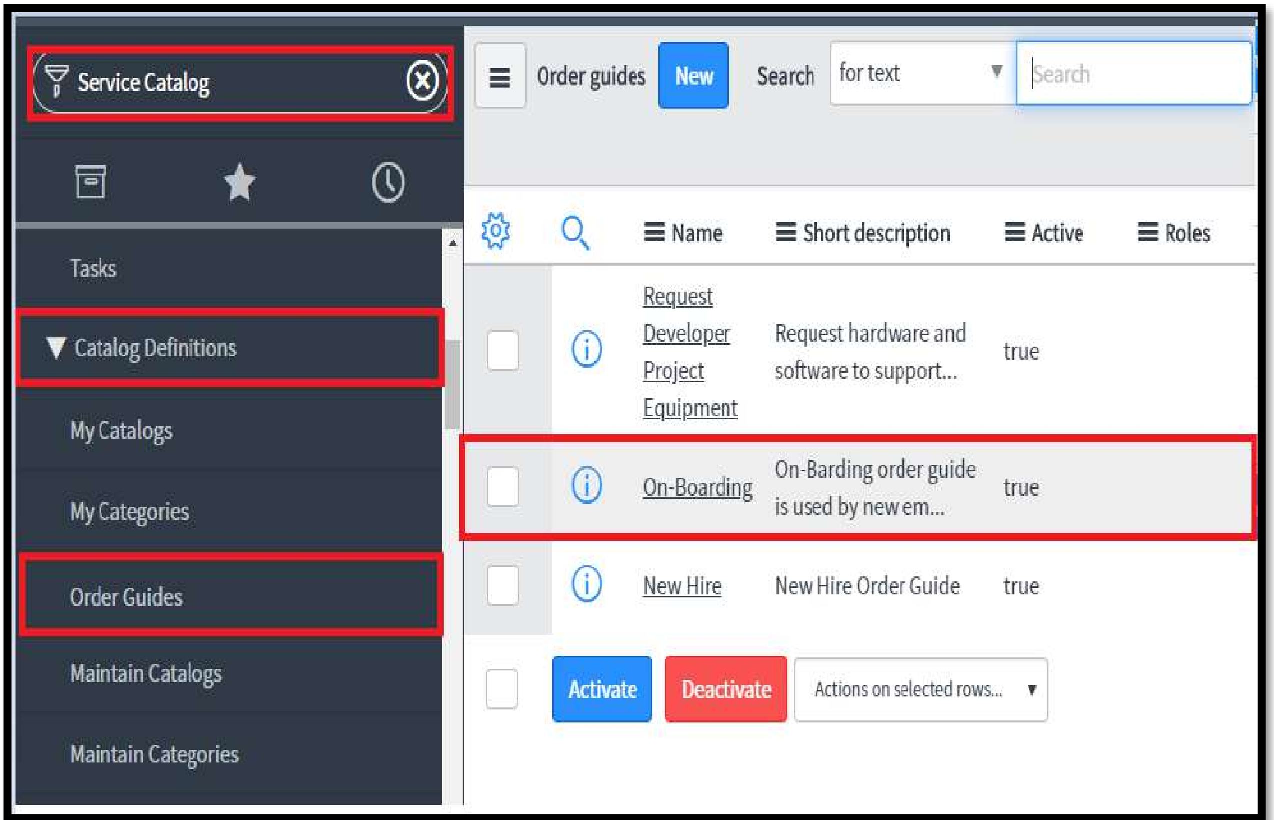
Example : A new employee **On-Boarding** order guide rule can define that if the **New Employee** Designation is **IT Technician or Senior Developer**, and the **Department is IT**, then we add an Apple Mac Pro 15 Laptop item to deployed to customer.

Example: A new employee need to do work from home, then we can include **VPN Connection**

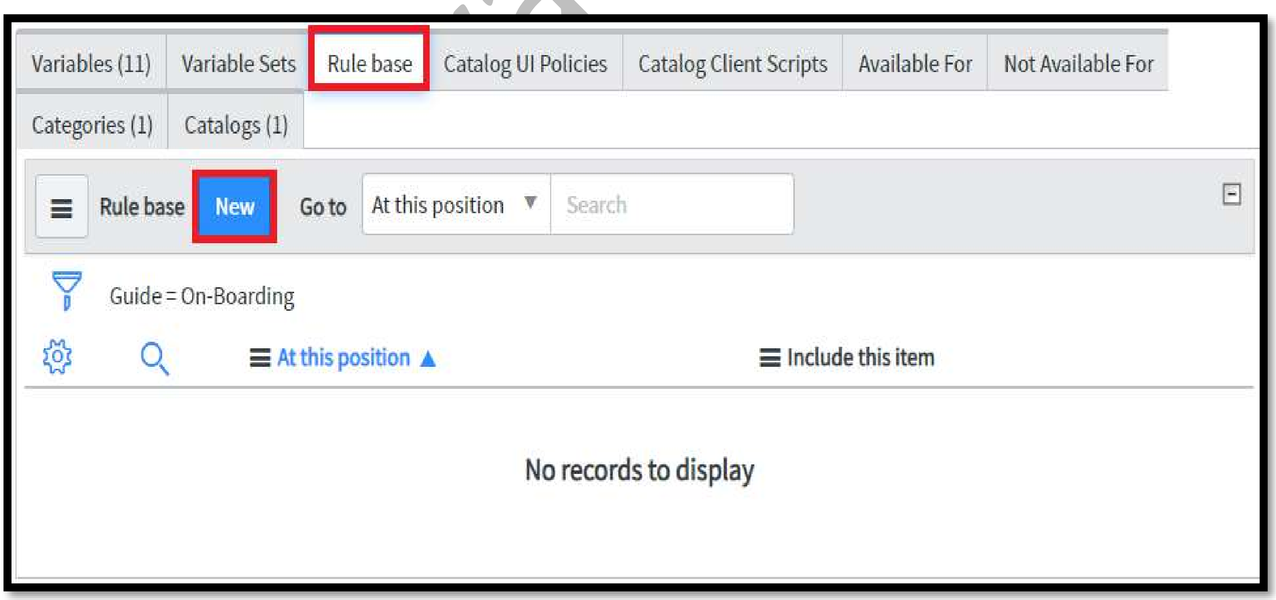
Rule: 1 Employee looking for work from home then include VPN Connection

Procedure

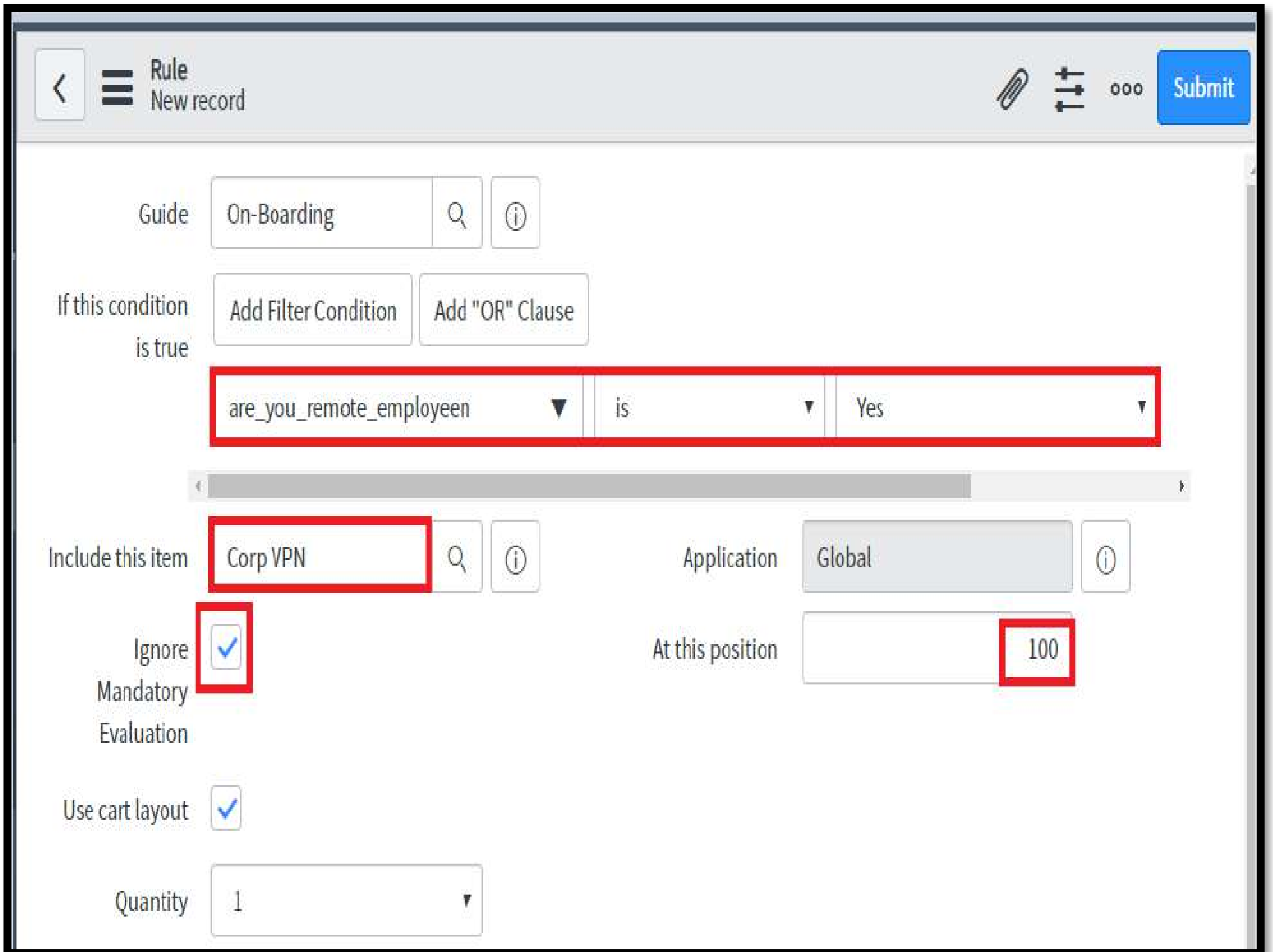
1. Navigate to **Service Catalog > Catalog Definition > Order Guides**
2. Open **On-Board** order guide.



3. Open **Rule Base** tab from related list,
4. Click on **New**

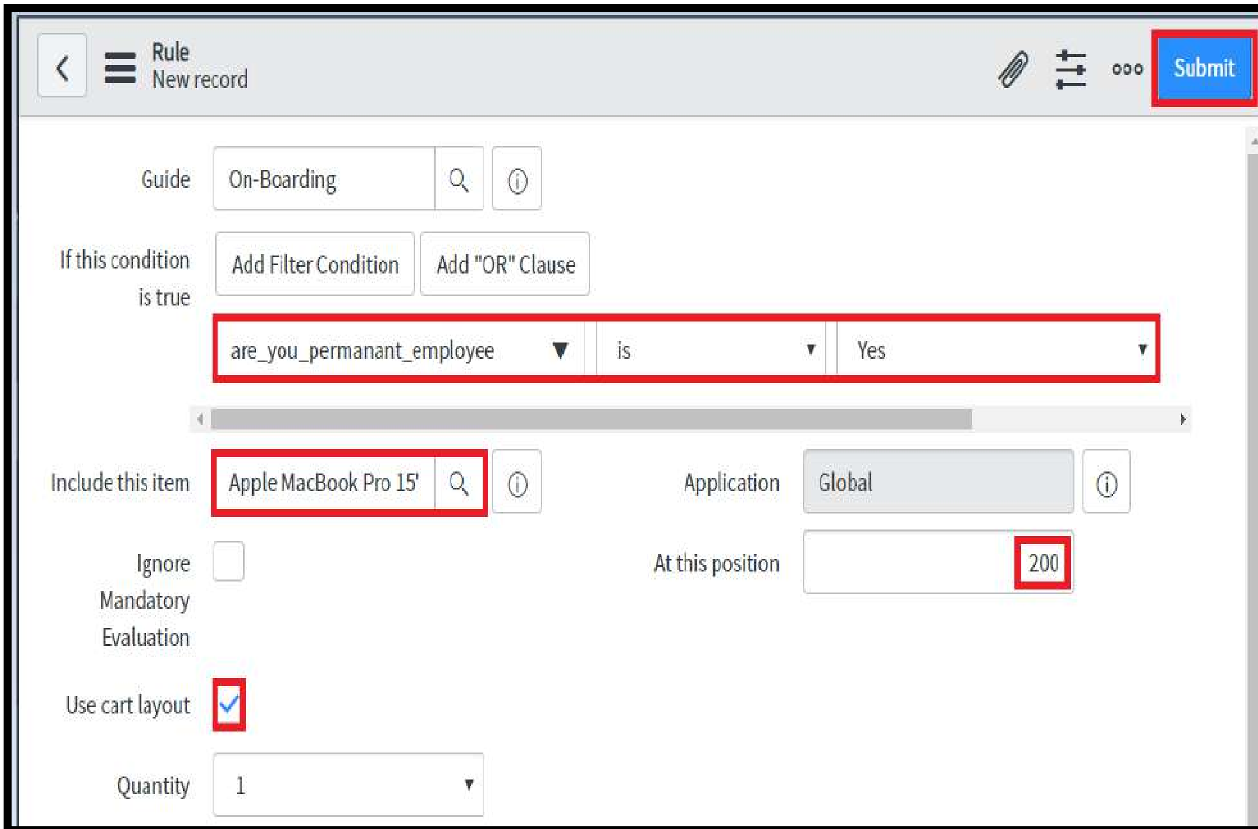


5. Fill **Rule Base** form



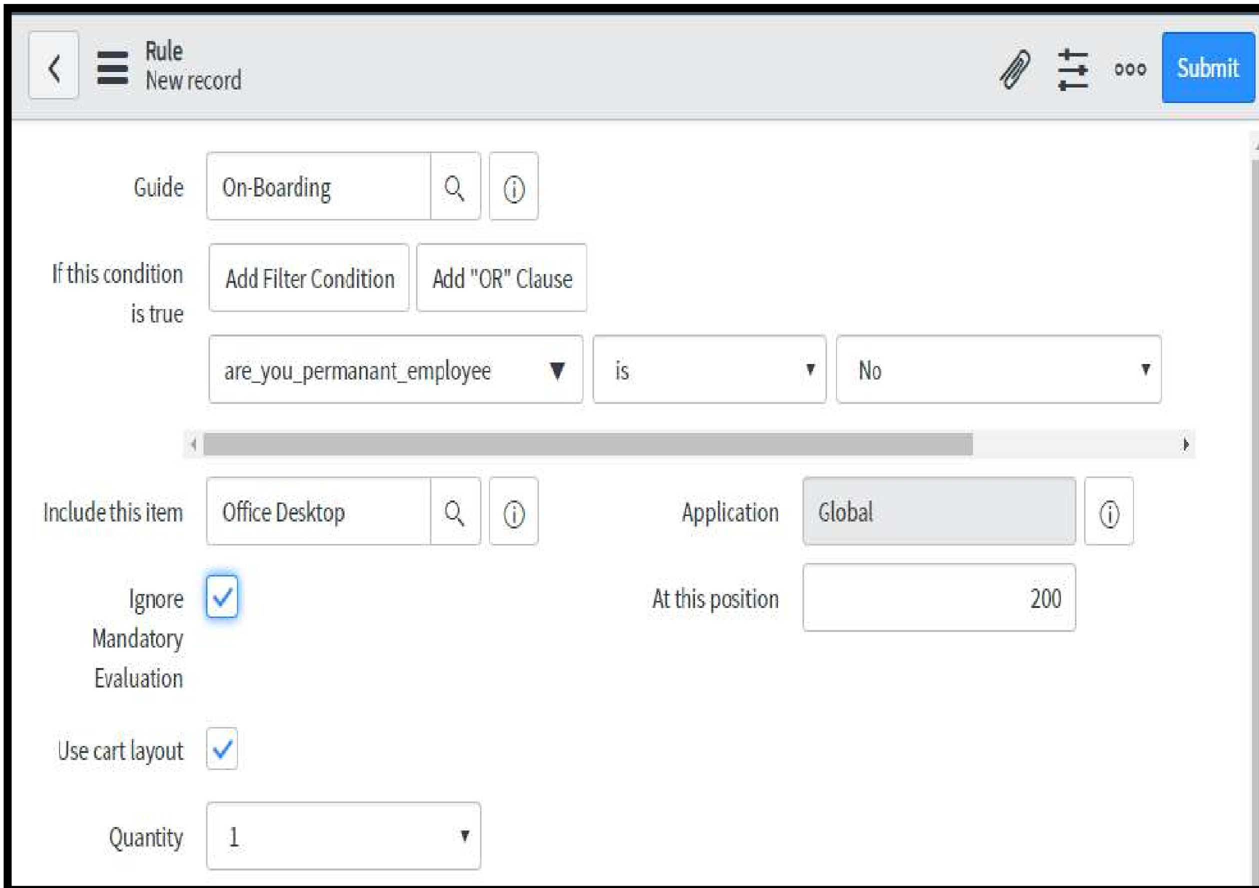
6. **Guide:** On-Boarding
7. **Condition:** [Remote employee] [is] [Yes]
8. **Include this item:** Corp VPN
9. **Ignore Mandatory Evolution:** Checked **True**
10. **Use cart layout:** Checked **True**
11. **At this position: 100**
12. Click on **Submit**

Rule: 2 Employee type is permanent is **Yes** Then including **Apple MacBook pro 15**



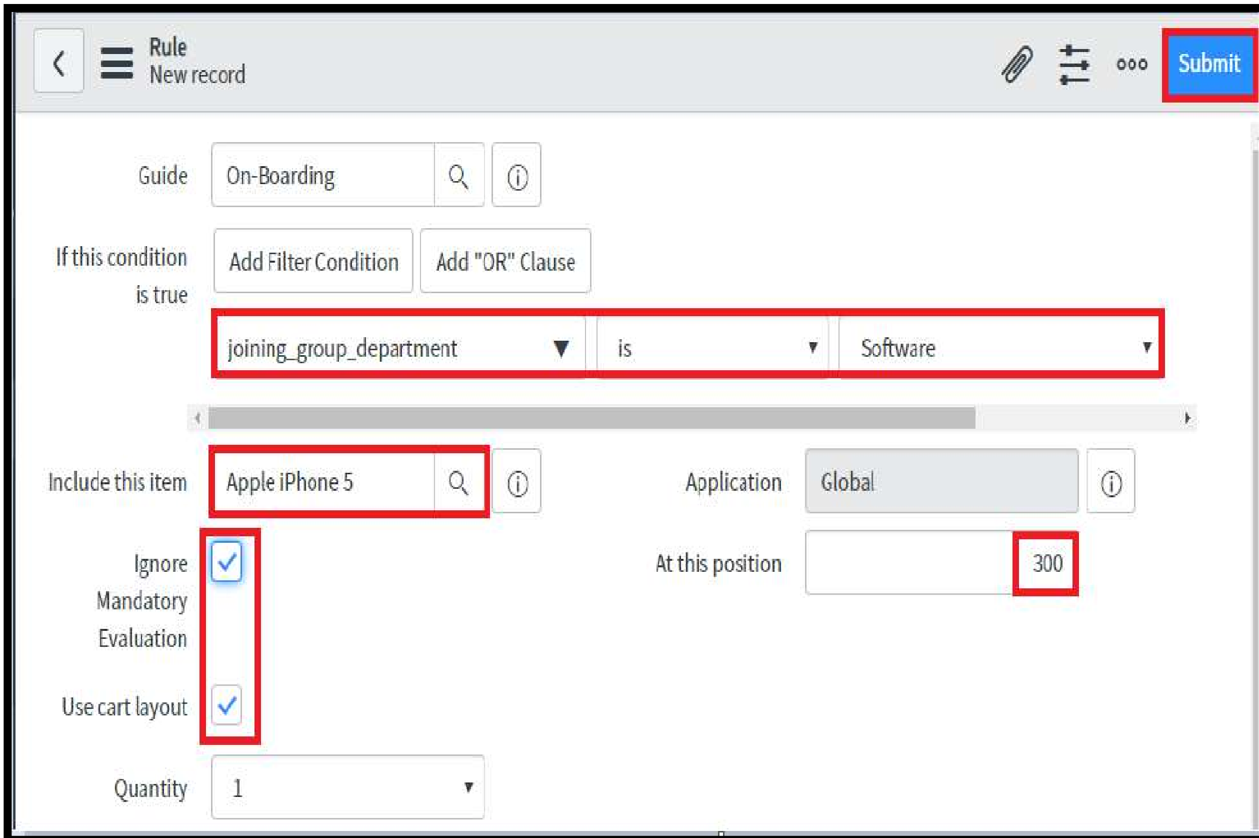
1. **Guide:** On-Boarding
2. **Condition:** [Permanent employee] [is] [Yes]
3. **Include this item:** Apple MacBook Pro15
4. **Ignore Mandatory Evolution:** Checked **True**
5. **Use cart layout:** Checked **True**
6. **At this position: 200**
7. Click on **Submit**

Rule: 3 Employee type is permanent is **No** Then including **Office Desktop**



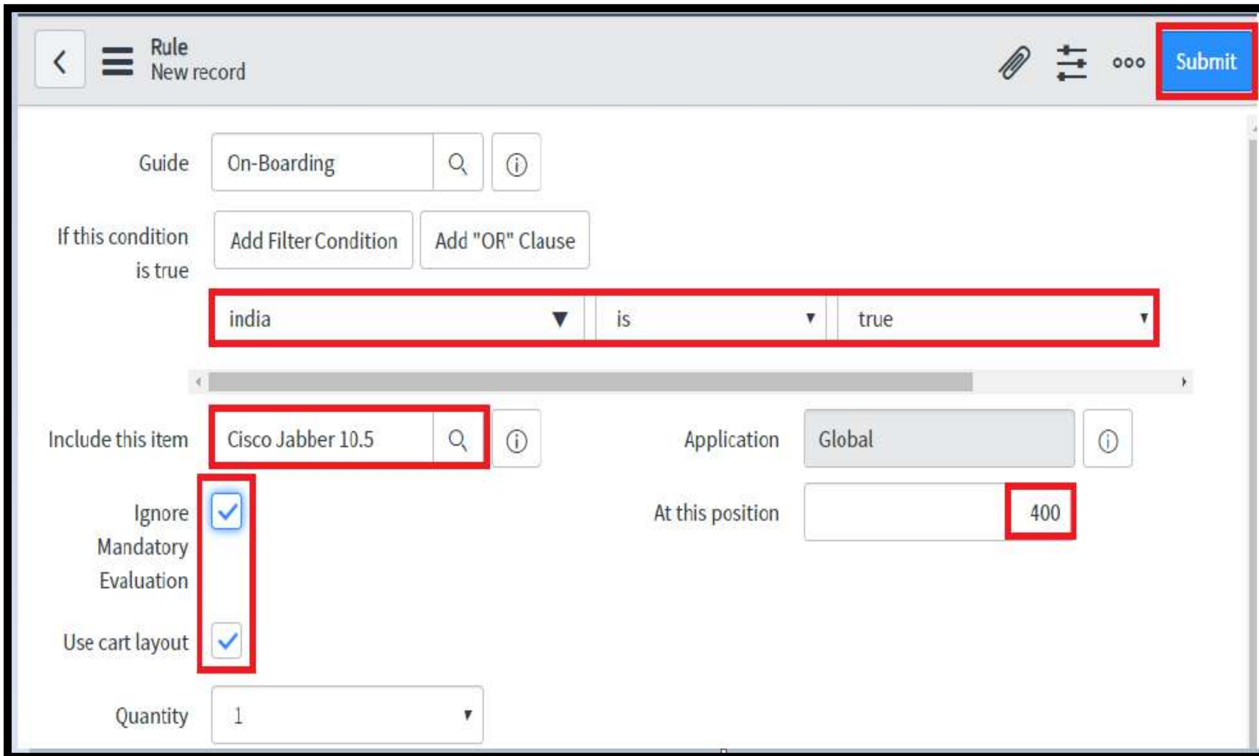
1. **Guide:** On-Boarding
2. **Condition:** [Permanent employee] [is] [No]
3. **Include this item:** Office Desktop
4. **Ignore Mandatory Evolution:** Checked **True**
5. **Use cart layout:** Checked **True**
6. **At this position: 200**
7. Click on **Submit**

Rule: 4 Employee Joining group department is Software **include Apple I phone 5**



1. **Guide:** On-Boarding
2. **Condition:** [Joining Group Department] [is] [Software]
3. **Include this item:** Apple IPhone 5
4. **Ignore Mandatory Evolution:** Checked **True**
5. **Use cart layout:** Checked **True**
6. **At this position: 300**
7. Click on **Submit**

Rule: 5 Employee Location is India Then including **Cisco Jabber 10.5**



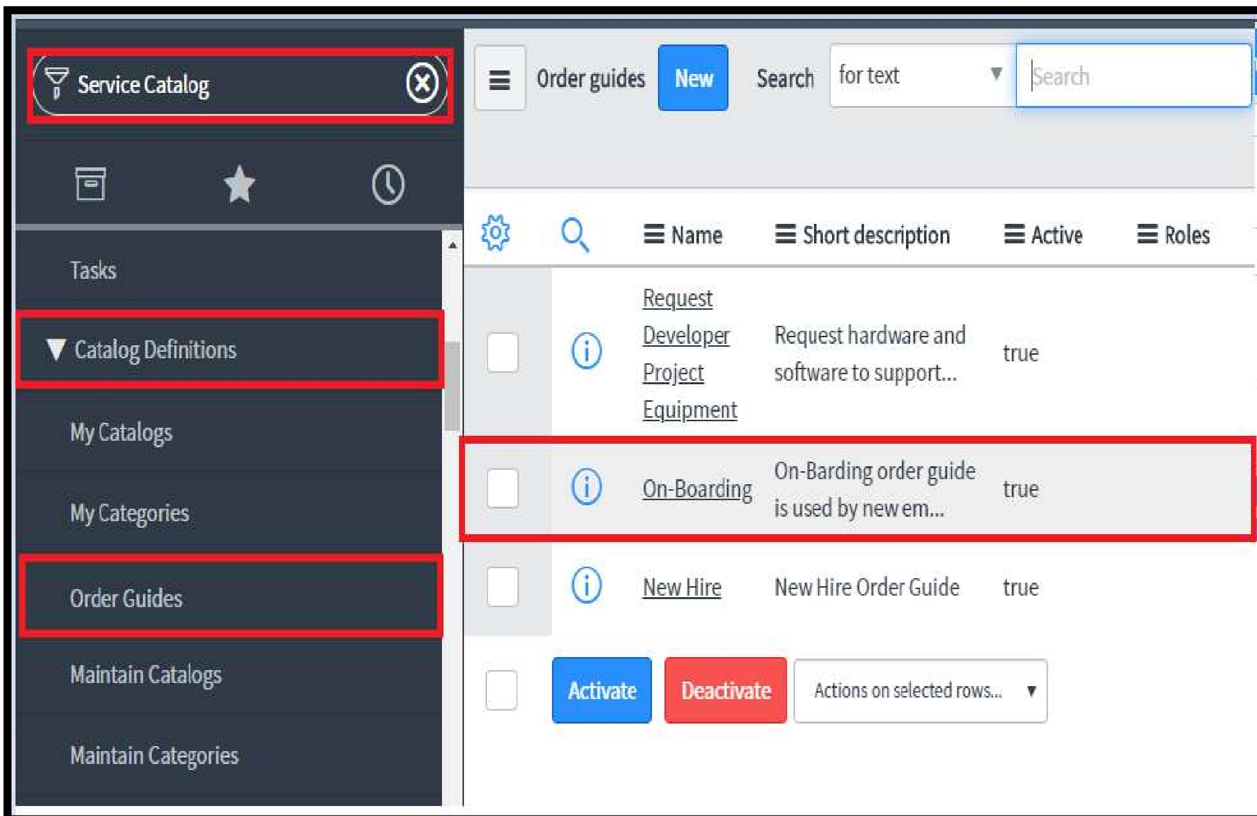
1. **Guide:** On-Boarding
2. **Condition:** [Joining Location] [India] [True]
3. **Include this item:** Cisco Jabber 10.5
4. **Ignore Mandatory Evolution:** Checked **True**
5. **Use cart layout:** Checked **True**
6. **At this position: 400**
7. Click on **Submit**

Exercise: 9 Order Guide task execution

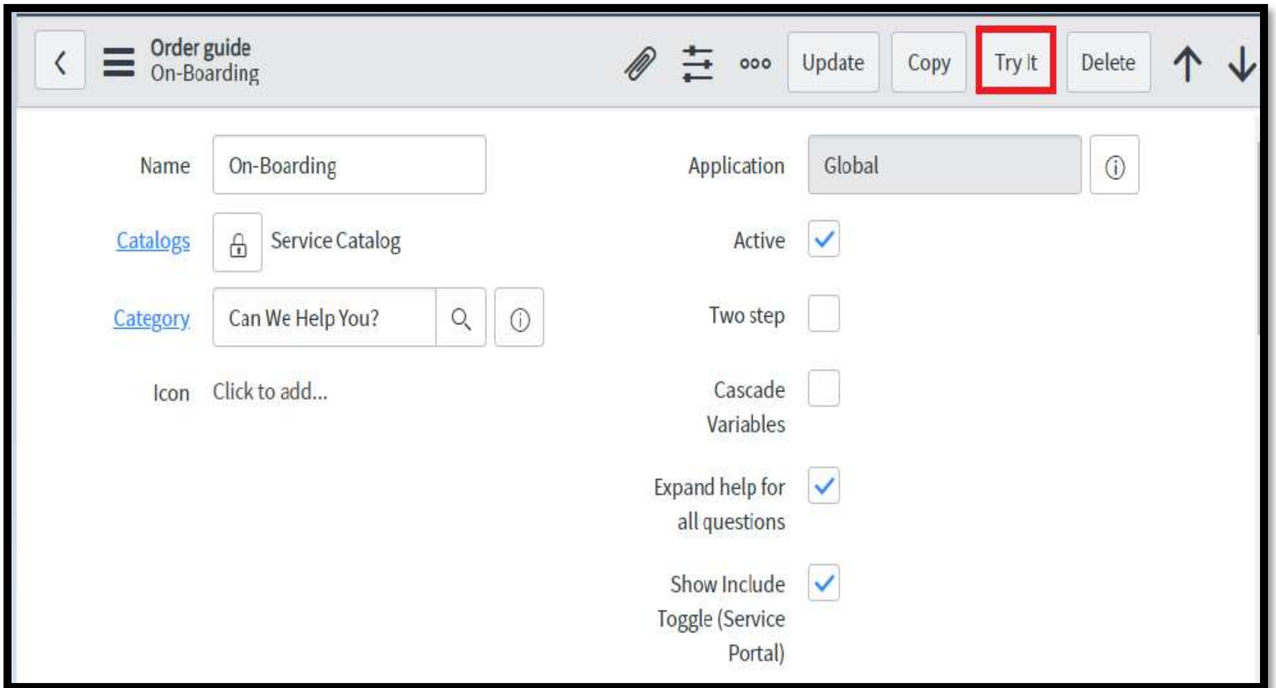
Fill all mandatory fields in **Order Guide** to requesting items

Procedure

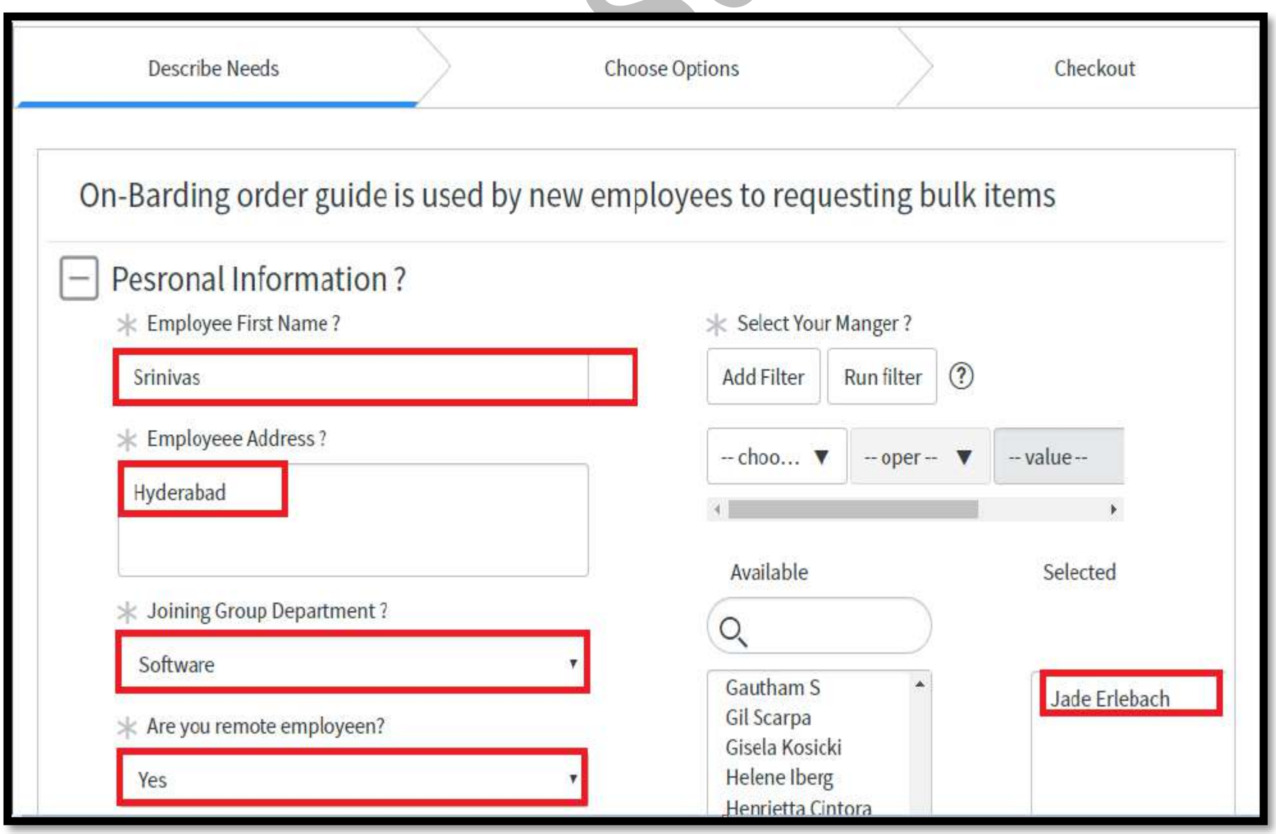
1. Navigate to **Service Catalog > Catalog Definition > Order Guides**
2. Open **On-Board** order guide



3. Click on **Try It** Button

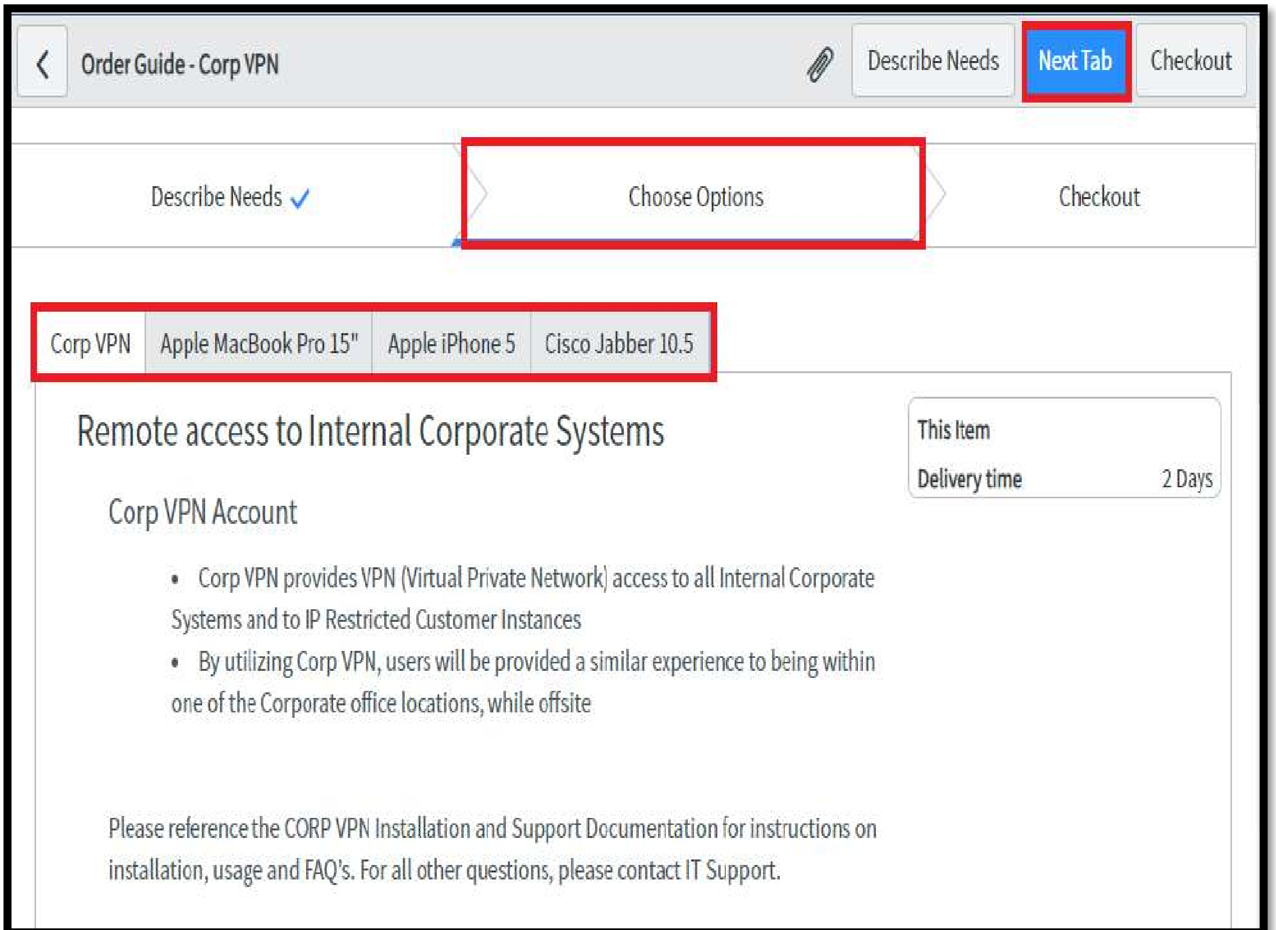


4. Fill All mandatory fields in **Order Guide**

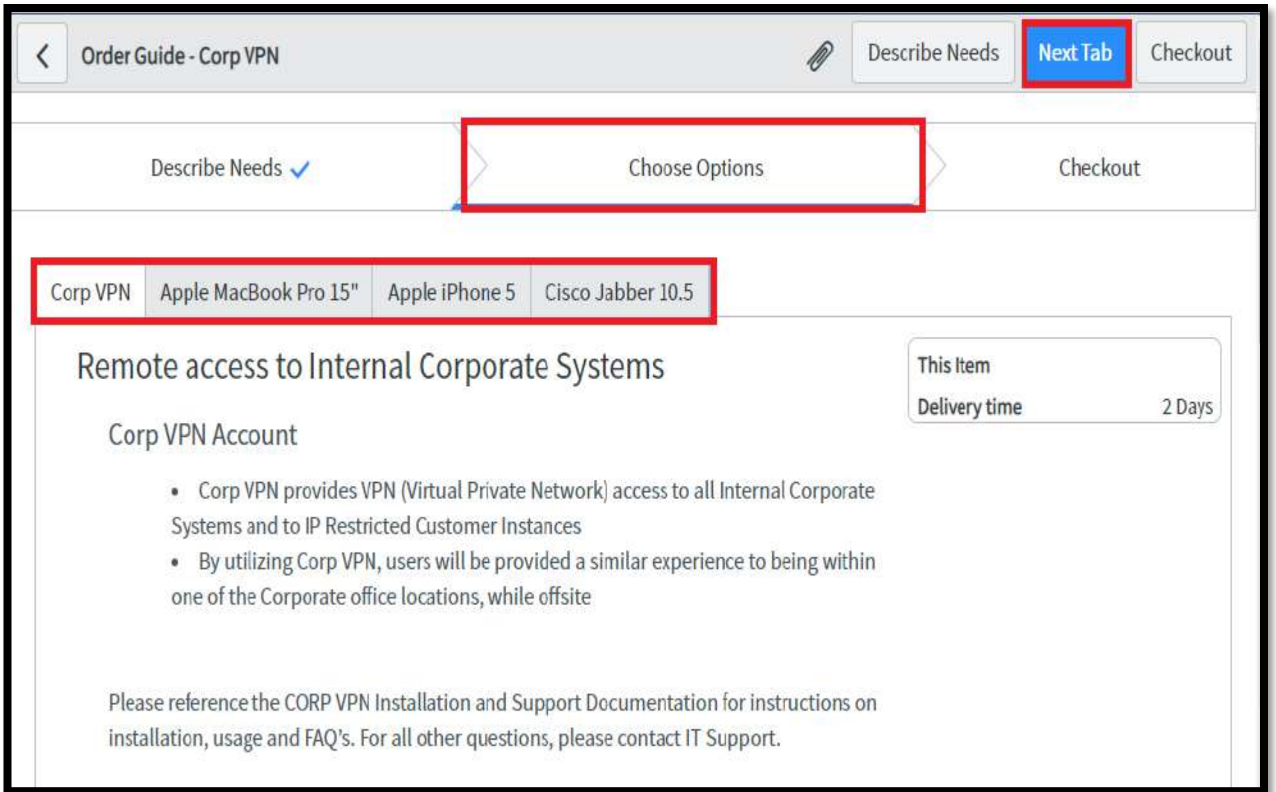


5. **Employee First Name**: Srinivas

6. **Employee Address**: Hyderabad
7. **Joining Group Department: Software**
8. **Are you remote employee: Yes**
9. **Select your manager:** Fred Luddy
10. **Location:** India
11. **Are your permanent employee:** Yes?
12. Click on **Choose Option**

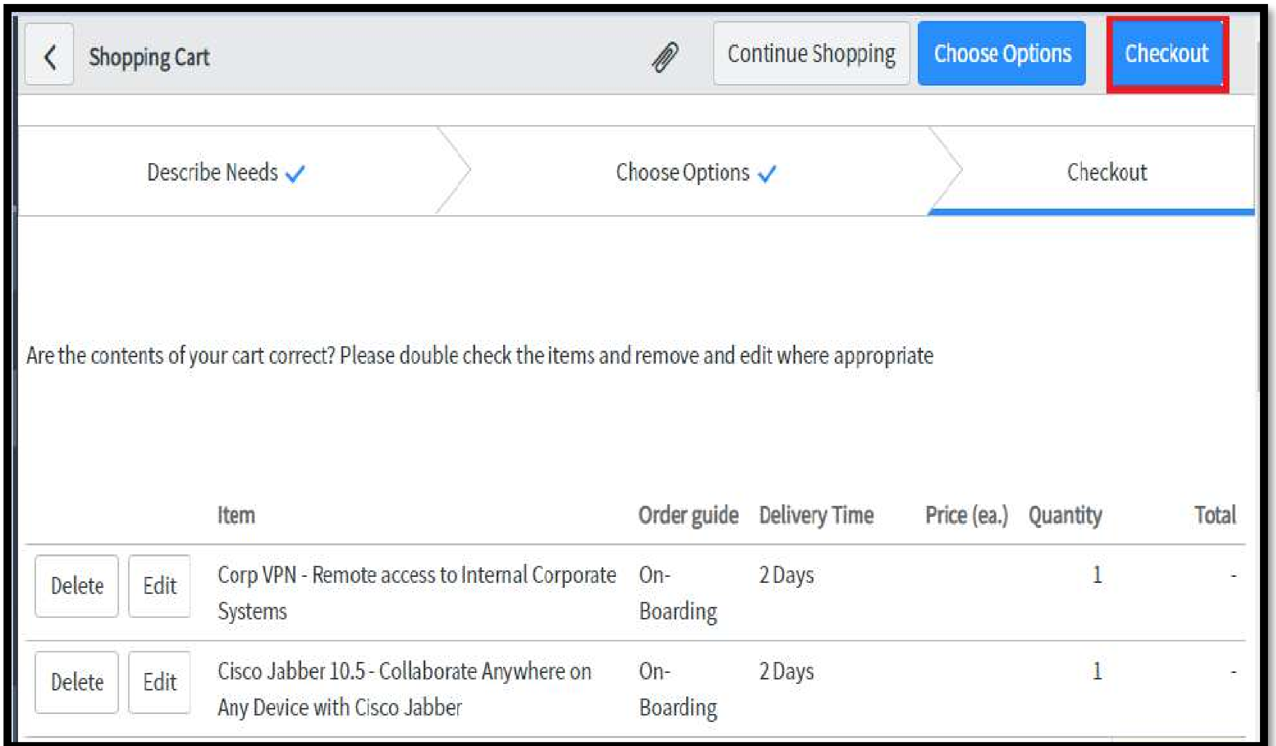


13. This employee eligibility to get these all items based on **Rules Base** configuration
14. Click on **Next Tab**

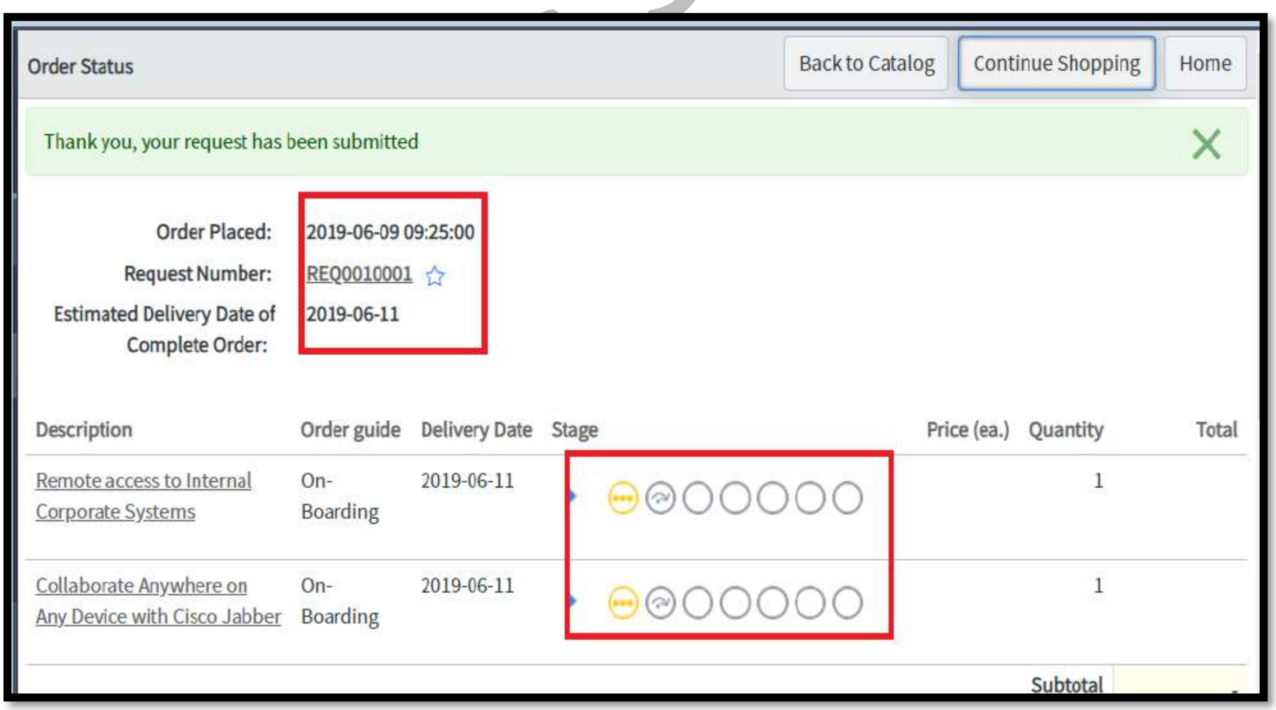


15. Click on **Checkout**

16. Again Click on **Checkout**



17. Below screen can define about your request from **On-Boarding** order guide



Working with Variable Sets

Variable sets can define collection variables create once and use multiple catalog items and record producers and order guides, it is for reusability purpose.

Ex: Service now catalog administrator defines 10 catalog items for same types of servers. The request process for all these items asks the same 10 questions, using the same variables.

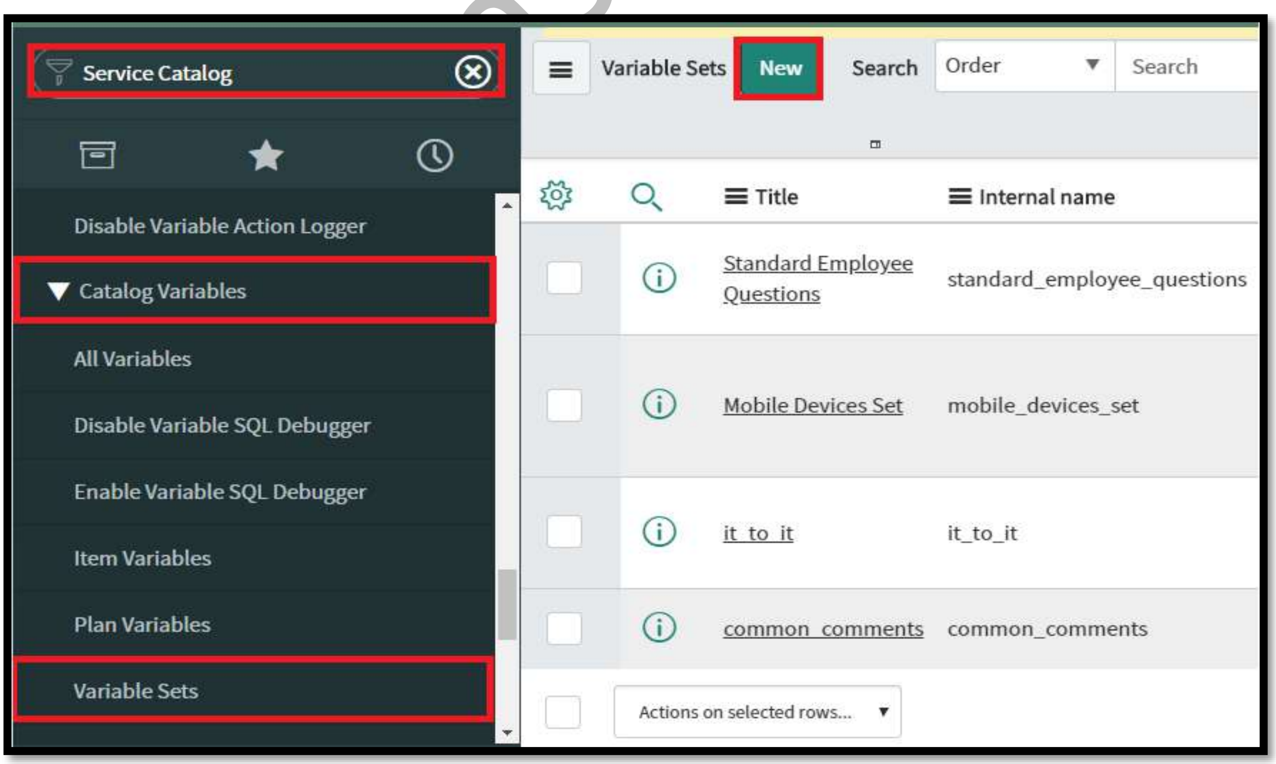
Associating these variables individually per catalog item is repetitive, time-consuming, and error-prone. Also, to make a single change to multiple catalog items involves manually changing each item. For example, to add a new variable to 10 catalog items, you would need to manually associate this new variable with each item.

Exercise: 10 Creating Variable Sets

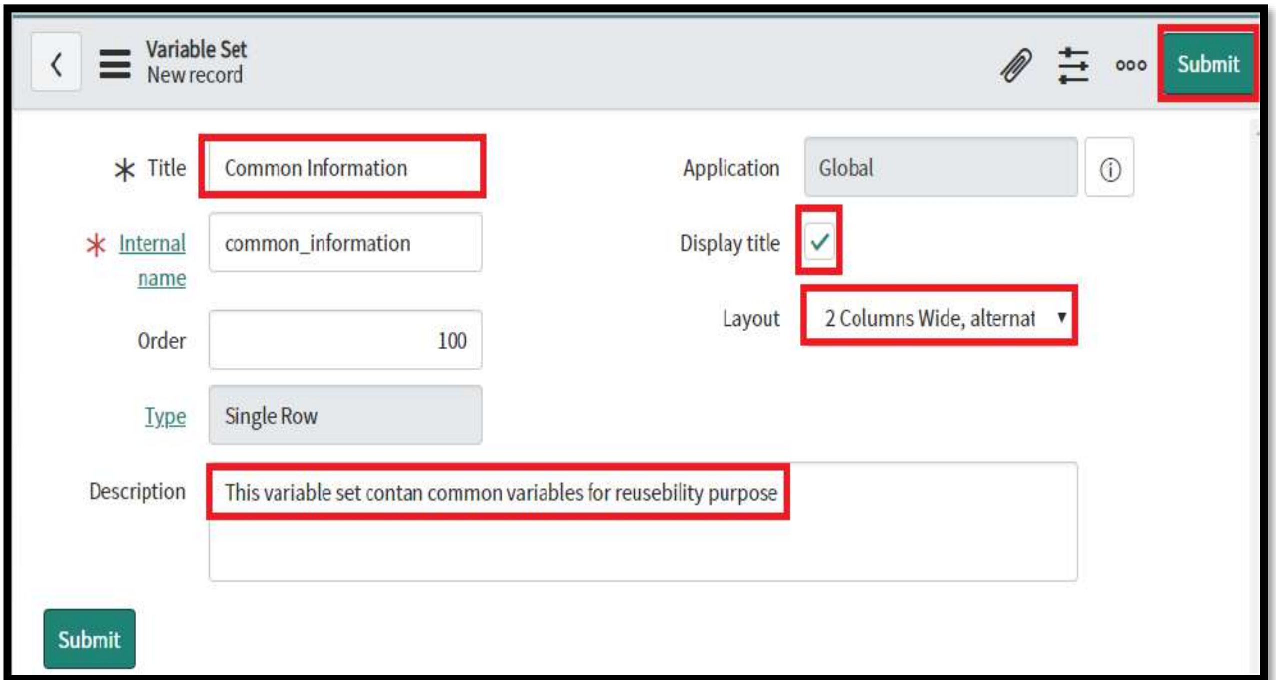
We can create a set of variables and then add them to multiple catalog items and order guides for reusability.

Procedure

1. Navigate to **Service Catalog > Catalog Variables > Variable Sets**
2. Click **New**.



3. Fill the **Variable Sets** form



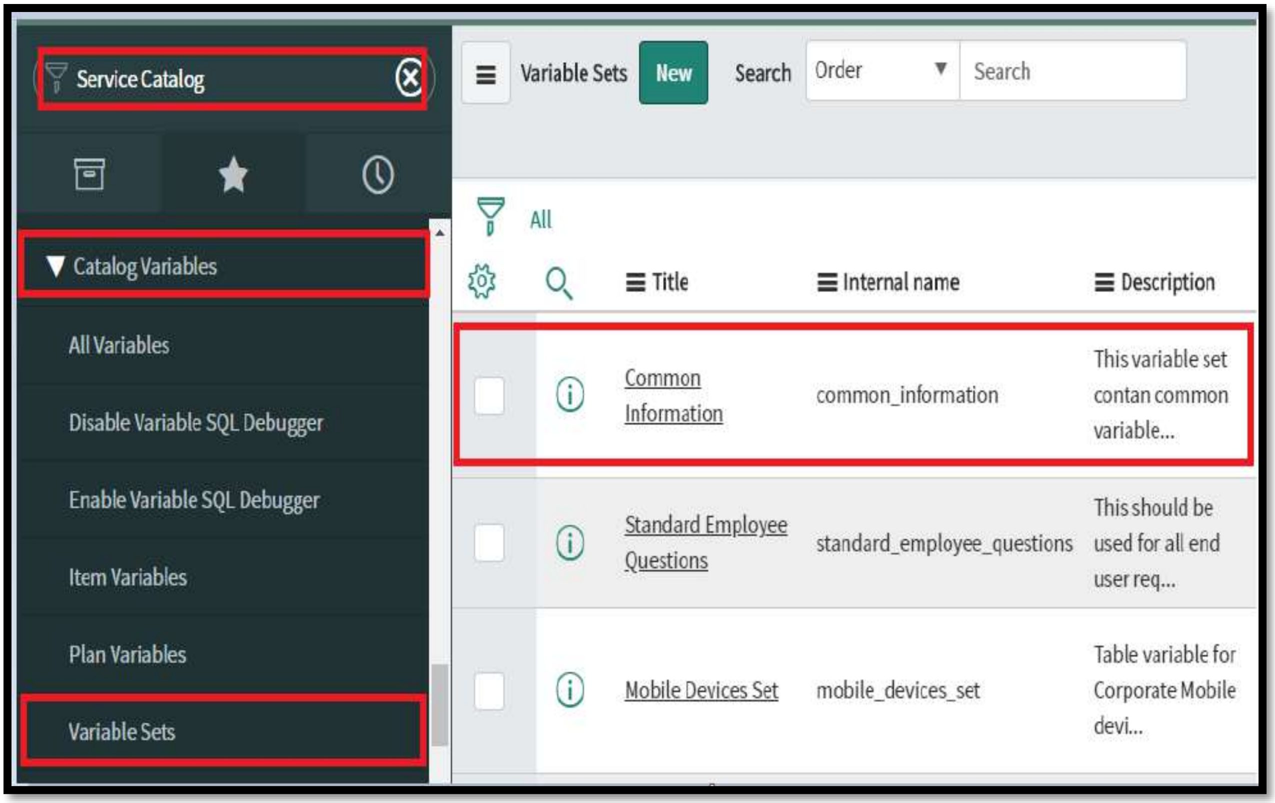
4. **Title:** Common Information
5. **Internal Name:** Common_Information
6. **Order:** 100
7. **Display Title:** Checked **True**
8. **Type:** Single Row
9. **Description: This** variable set is contain common variables for reusability purpose
10. Click on **Submit**

Working with Single Line Text Variable

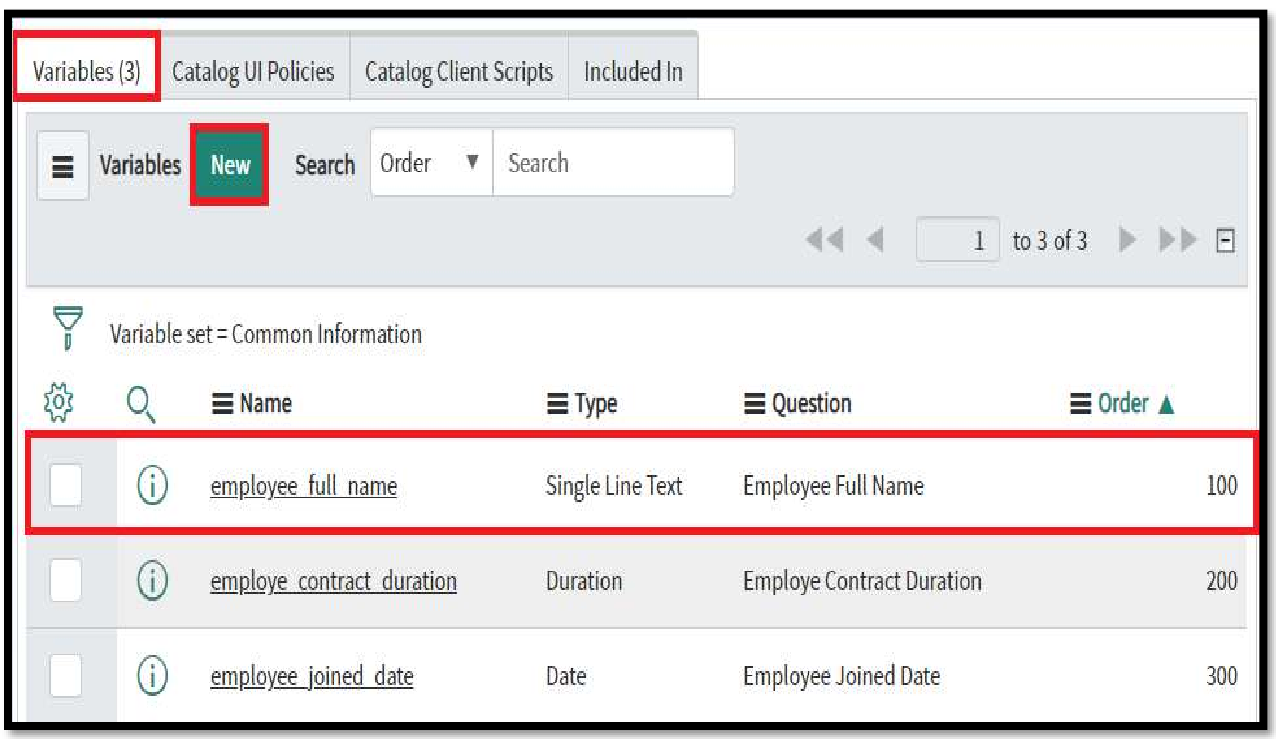
This variable is used to enter single line text value

Procedure

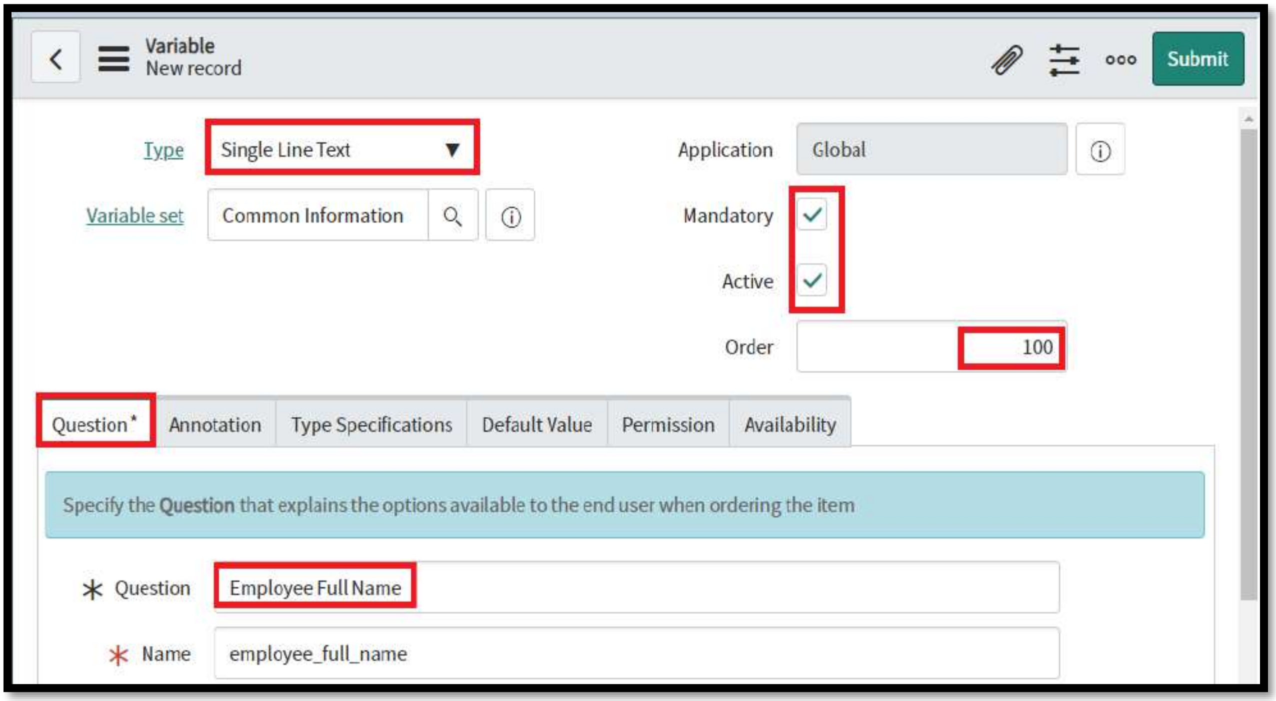
1. Navigate to **Service Catalog** > **Catalog Variables** > **Variable Sets > Common Information**



2. Open **Variable Tab** from **Related list**
3. Click on **New**



4. Fill **Single Line Text** Variable form



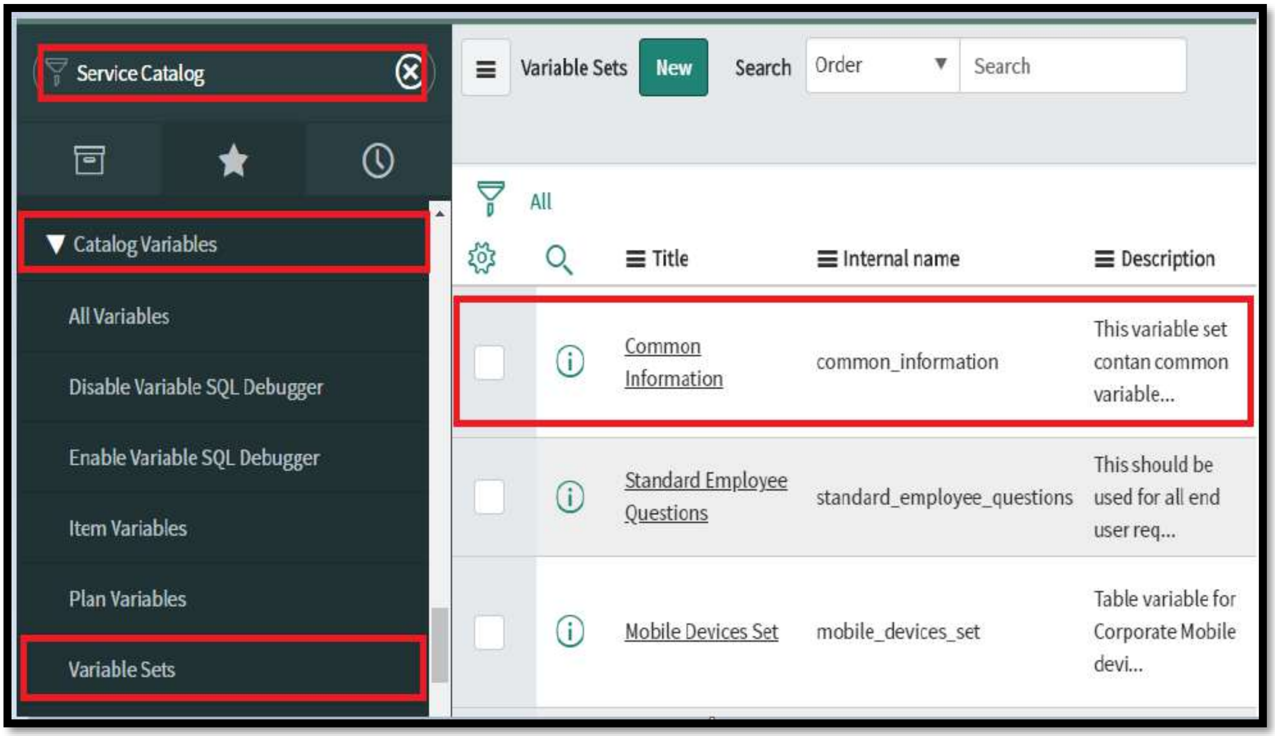
5. **Variable Type:** Single Line Text
6. **Mandatory:** Checked **True**
7. **Order:** 100
8. **Question:** Employee Full Name
9. **Name:** Auto Filled
10. Click on **Submit**

Working with Duration Variable

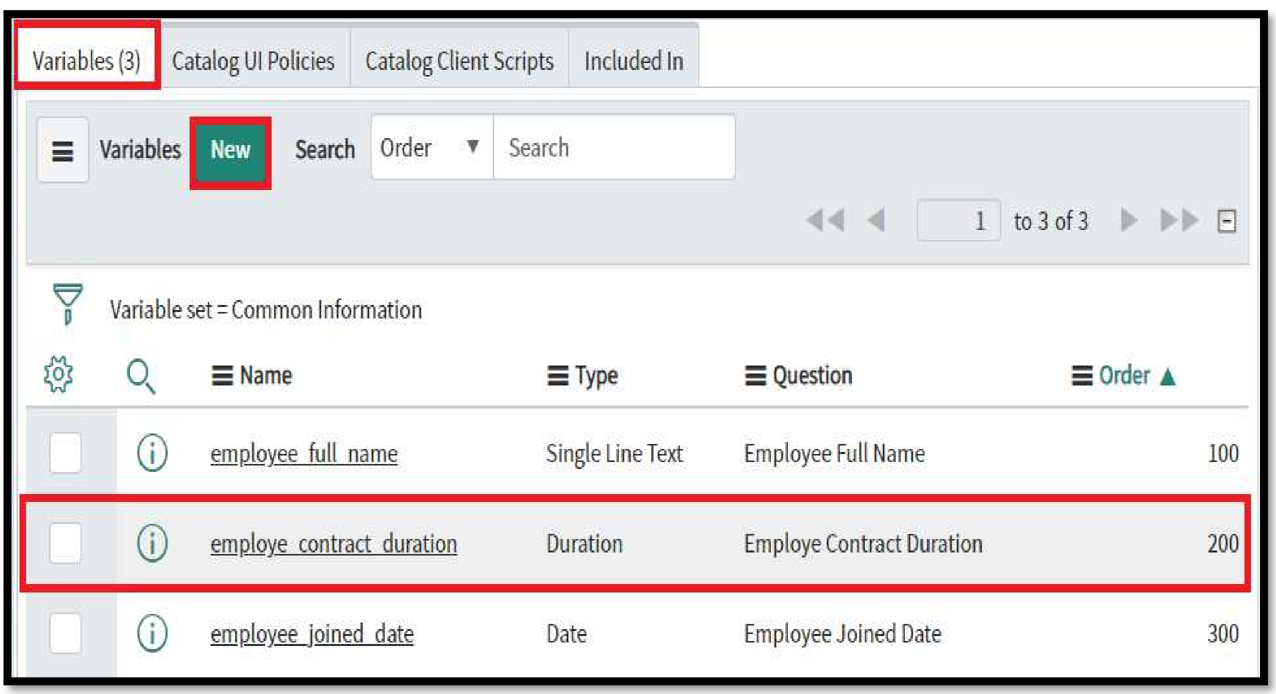
We use the duration variable to specify the duration for which a catalog item is available.

Procedure

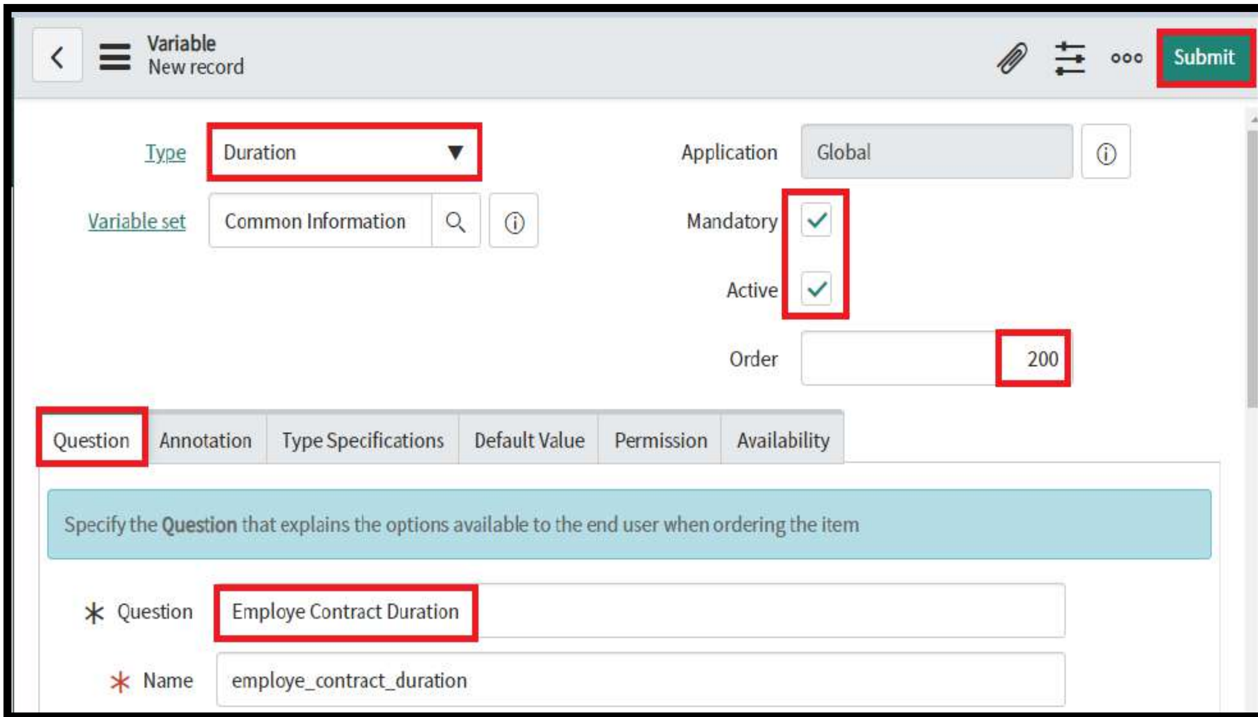
1. Navigate to **Service Catalog** > **Catalog Variables** > **Variable Sets > Common Information**



2. Open **Variable Tab** from **Related list**
3. Click on **New**



4. Fill single **Duration Variable** form



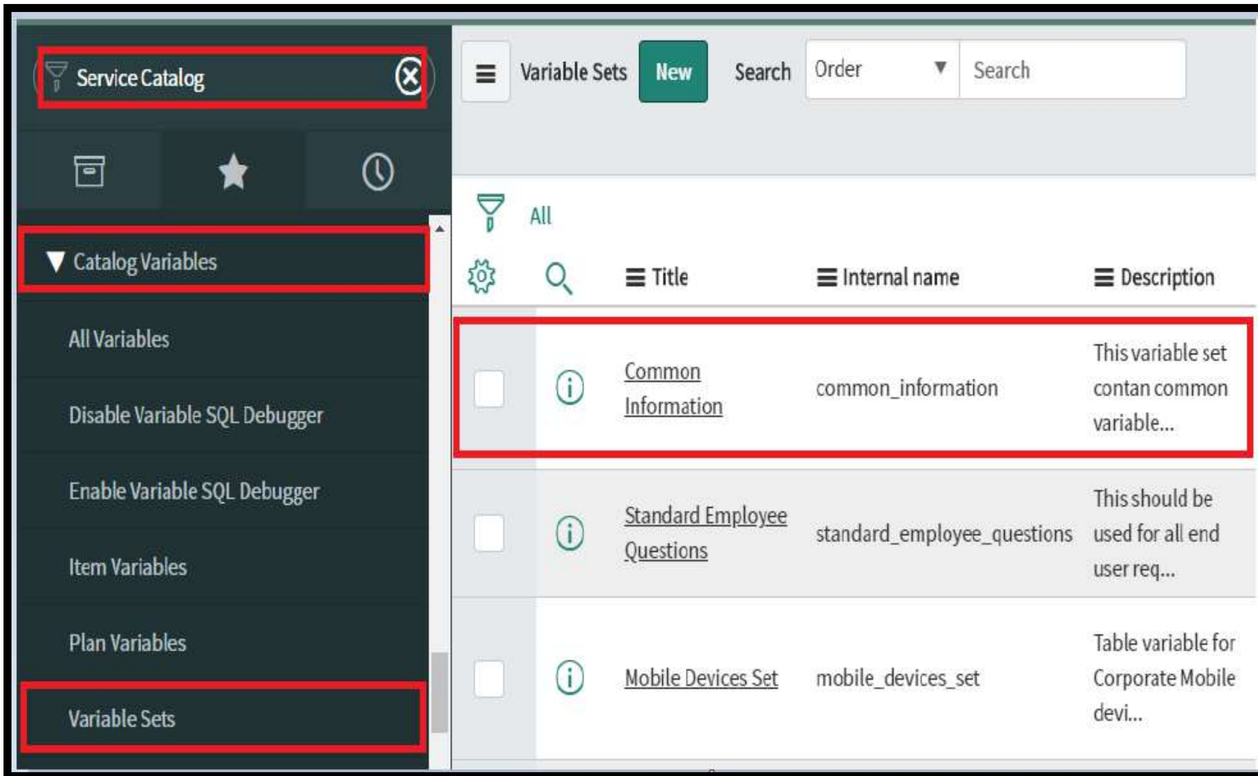
5. **Variable Type:** Duration
6. **Mandatory:** Checked **True**
7. **Order:** 200
8. **Question:** Employee Contract Duration
9. **Name:** Auto Filled
10. Click on **Submit**

Working with Date Variable

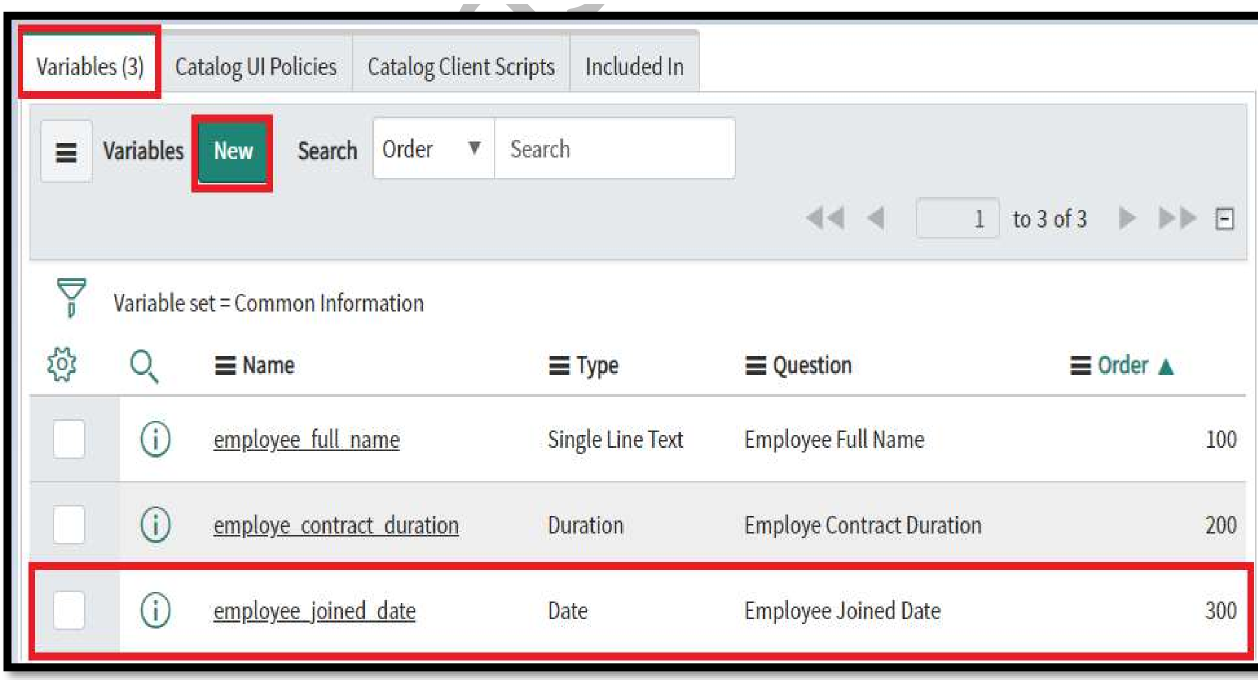
This variable is used select calendar date value.

Procedure

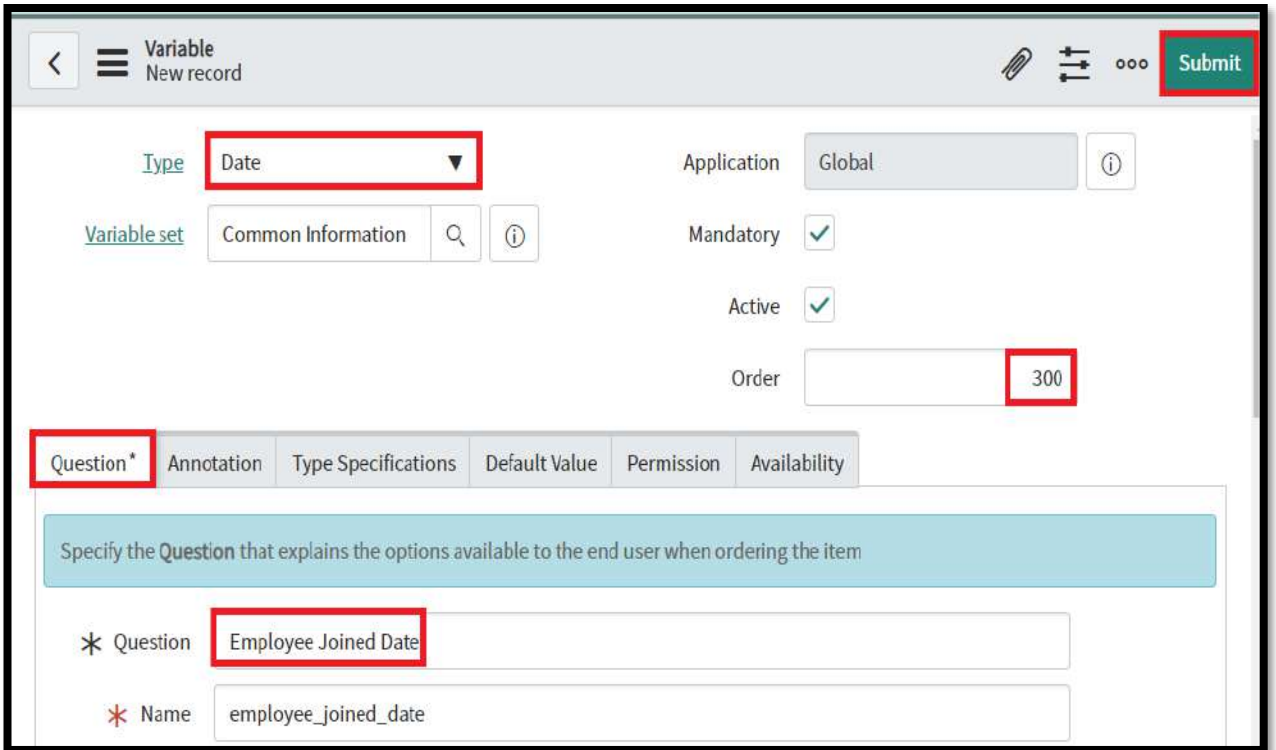
1. Navigate to **Service Catalog** > **Catalog Variables** > **Variable Sets > Common Information**



2. Open **Variable Tab** from **Related list**
3. Click on **New**



4. Fill single **Date Variable** form



5. **Variable Type:** Date
6. **Mandatory:** Checked **True**
7. **Order:** 300
8. **Question:** Employee Joined Date
9. **Name:** Auto Filled
10. Click on **Submit**

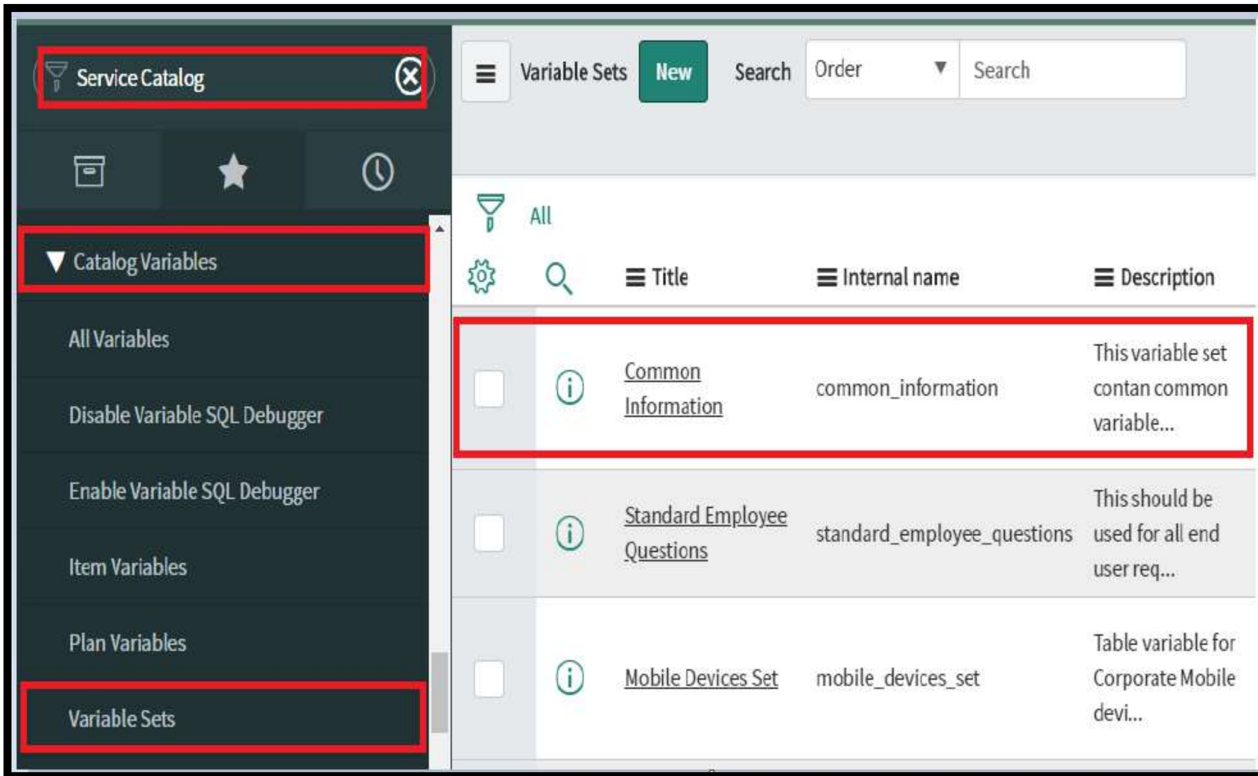
Working with Masked Variable

The masked variable inserts a field that masks the text entered. Each character entered is displayed as an asterisk **(*).** Use this variable to ask users for sensitive or confidential data such as passwords.

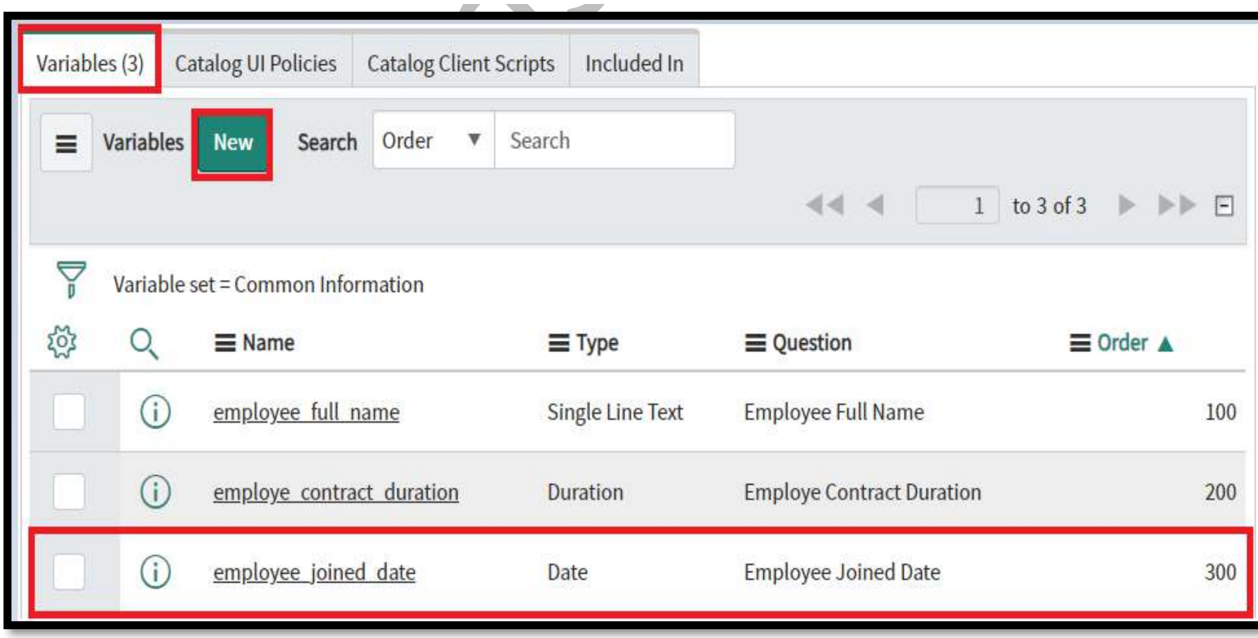
This variable is used get the value in encrypted format. These values can be decrypted by using

Procedure

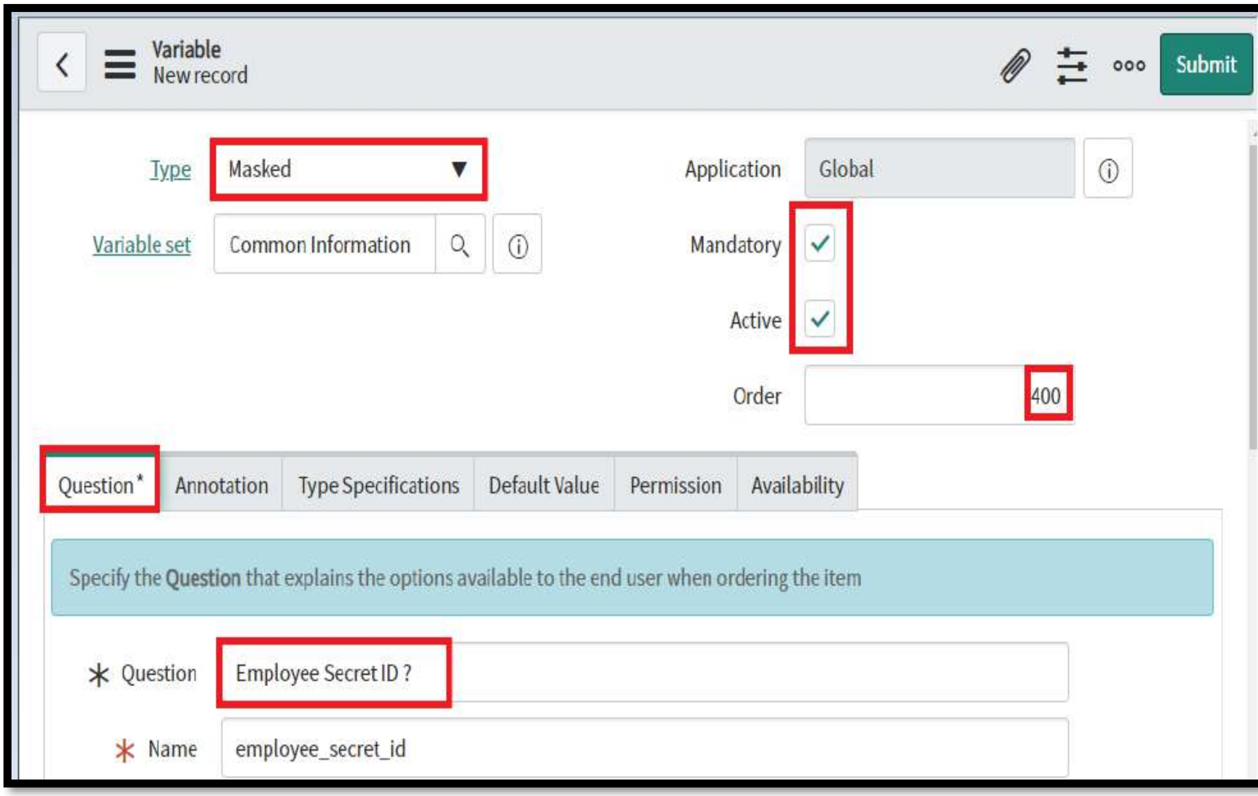
1. Navigate to **Service Catalog** > **Catalog Variables** > **Variable Sets > Common Information**



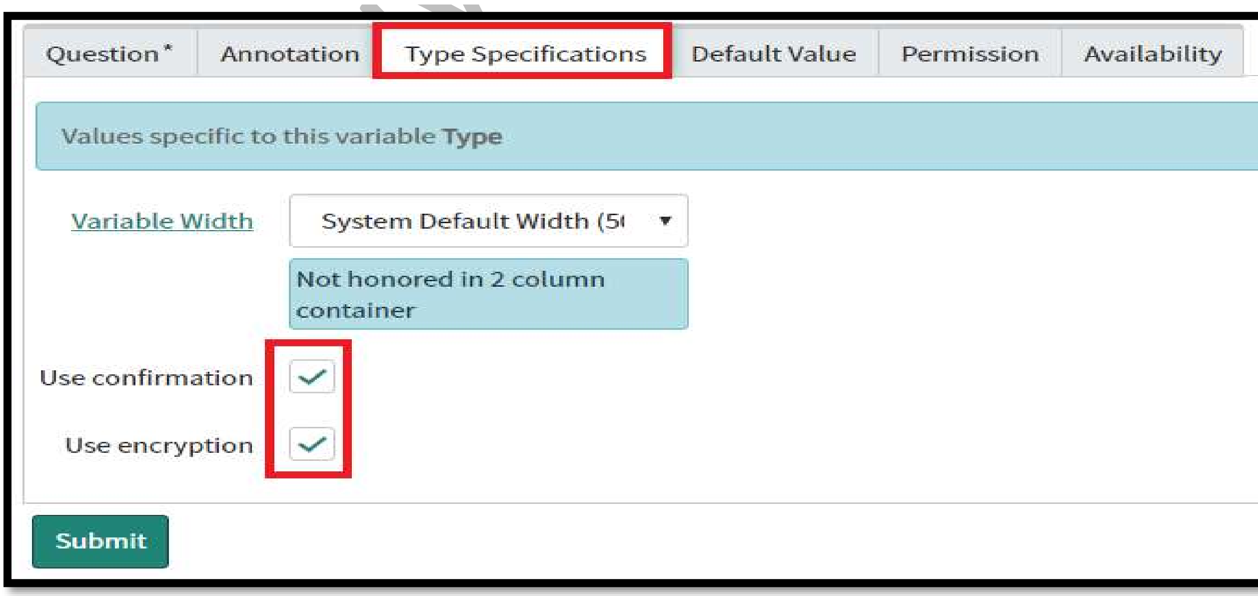
2. Open **Variable Tab** from **Related list**
3. Click on **New**



4. Fill single **Masked Variable** form



5. **Variable Type:** Masked
6. **Mandatory:** Checked **True**
7. **Order:** 400
8. **Question:** Employee Secret ID?
9. **Name:** Auto Filled



10. Open **Specification Tab**

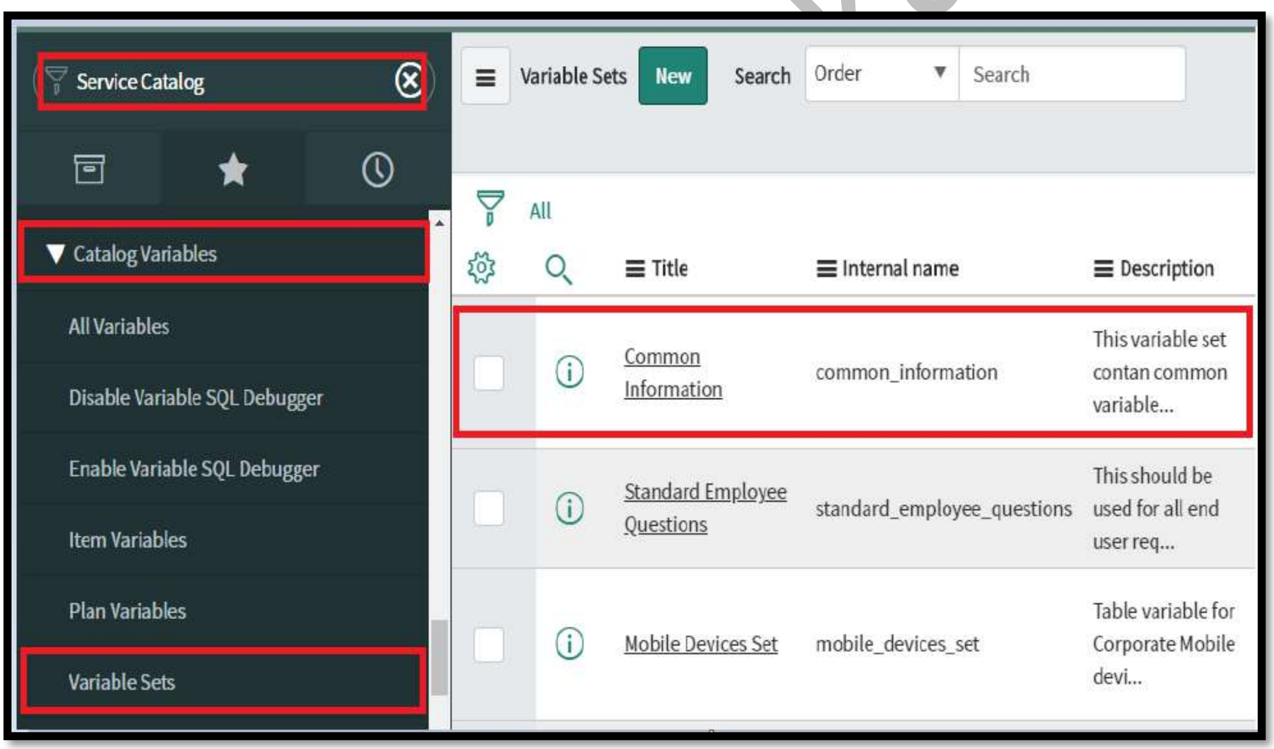
11. **Use Confirmation:** Checked True
12. **Use Encryption:** Checked True
13. Click on **Submit**

Working with HTML Variable

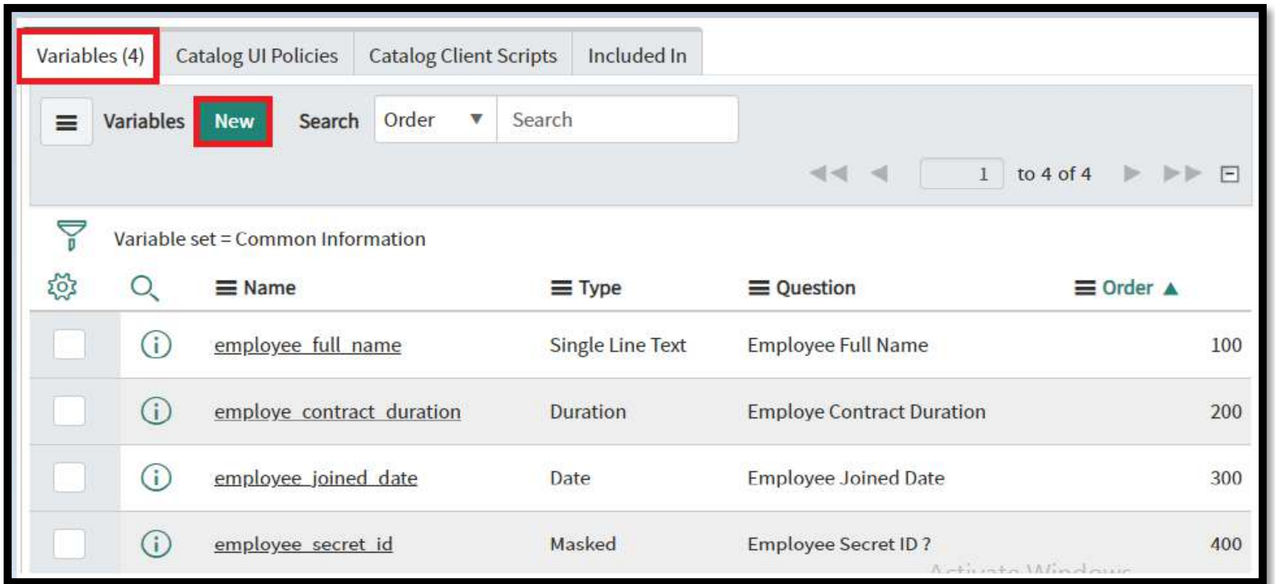
HTML Variable is used to get content in advanced format and Images, supporting for hyperlinks display content in editable mode

Procedure

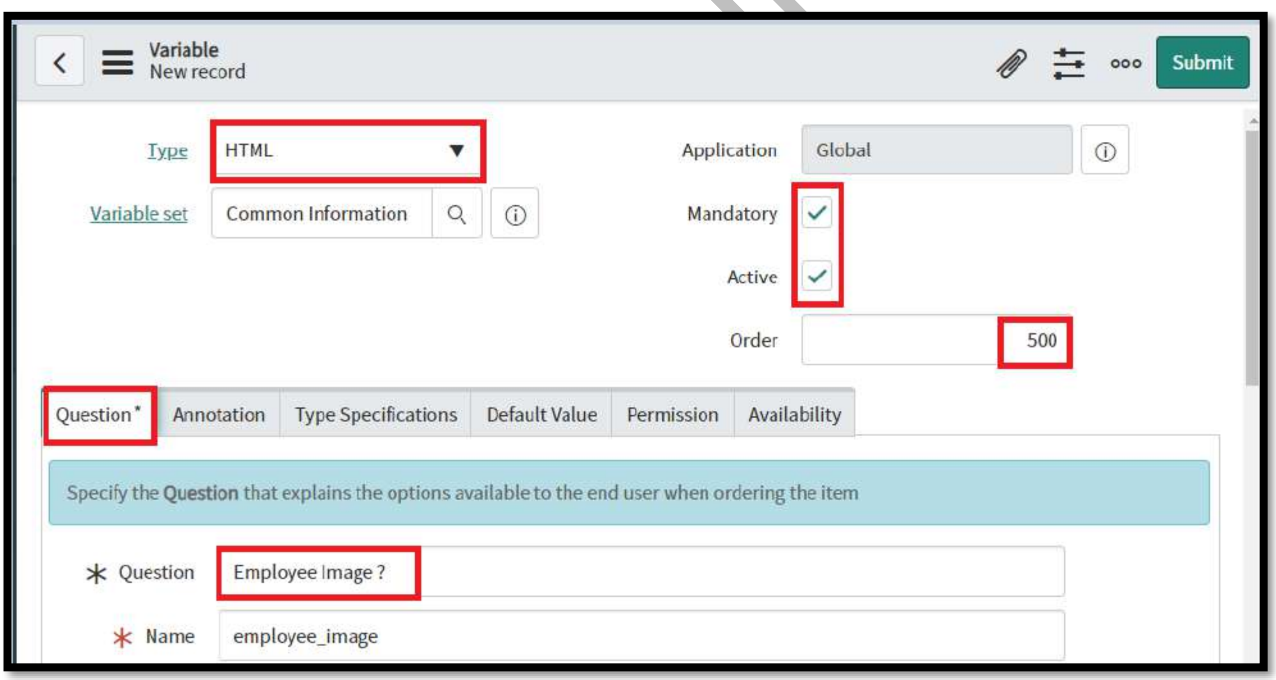
1. Navigate to **Service Catalog** > **Catalog Variables** > **Variable Sets > Common Information**



2. Open **Variable Tab** from **Related list**
3. Click on **New**



4. Fill **HTML Variable** form



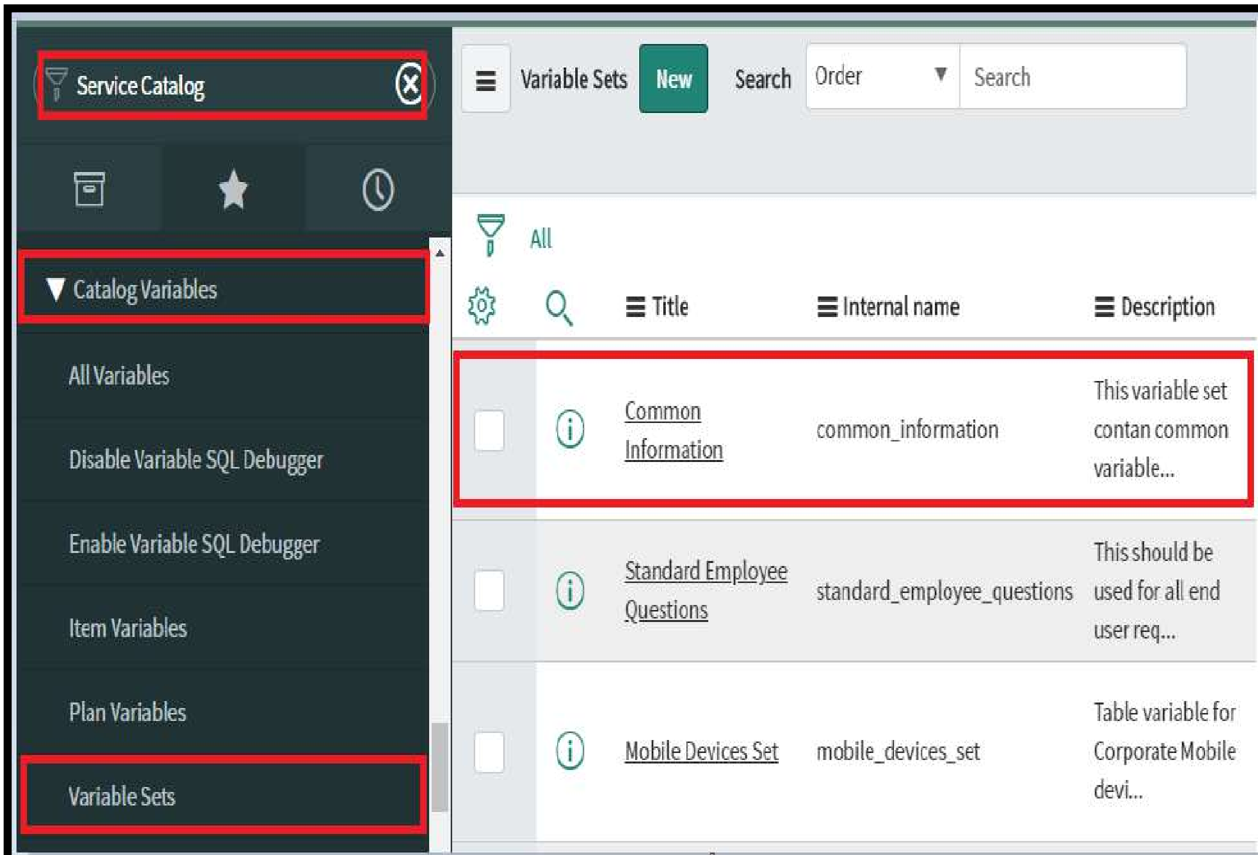
5. **Variable Type**: HTML
6. **Mandatory:** Checked True
7. **Order:** 500
8. **Question:** Employee Image?
9. **Name:** Auto Filled
10. Click on **Submit**

Working with Wide Single Line Text Variable

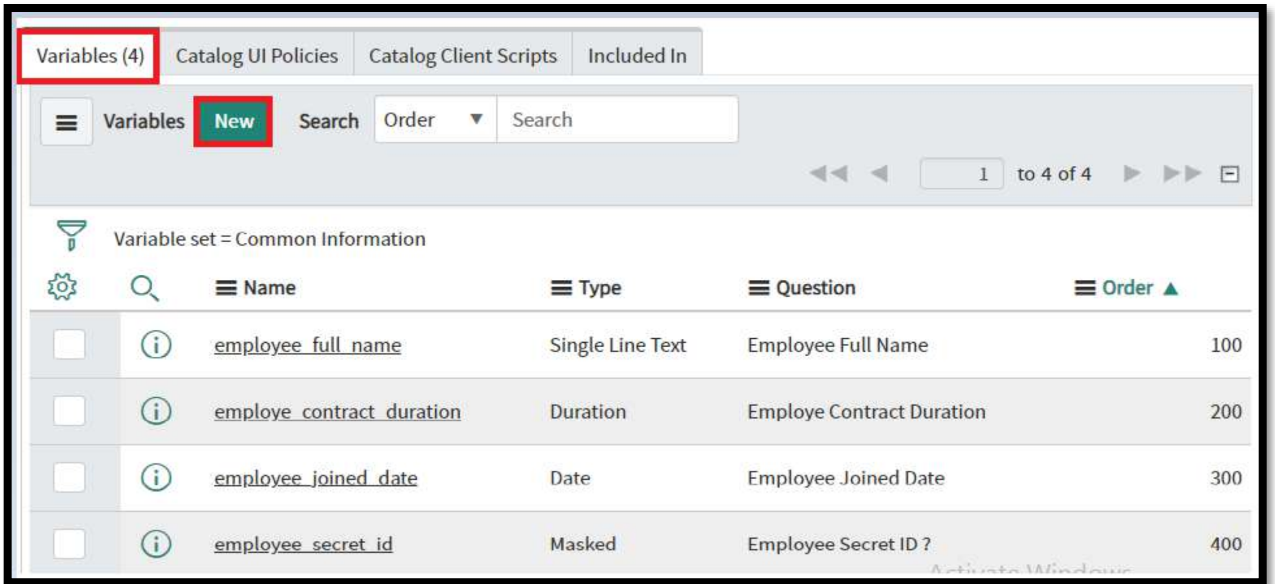
The **Wide Single Line Text** is used to allowing customer to enter longer line text value

Procedure

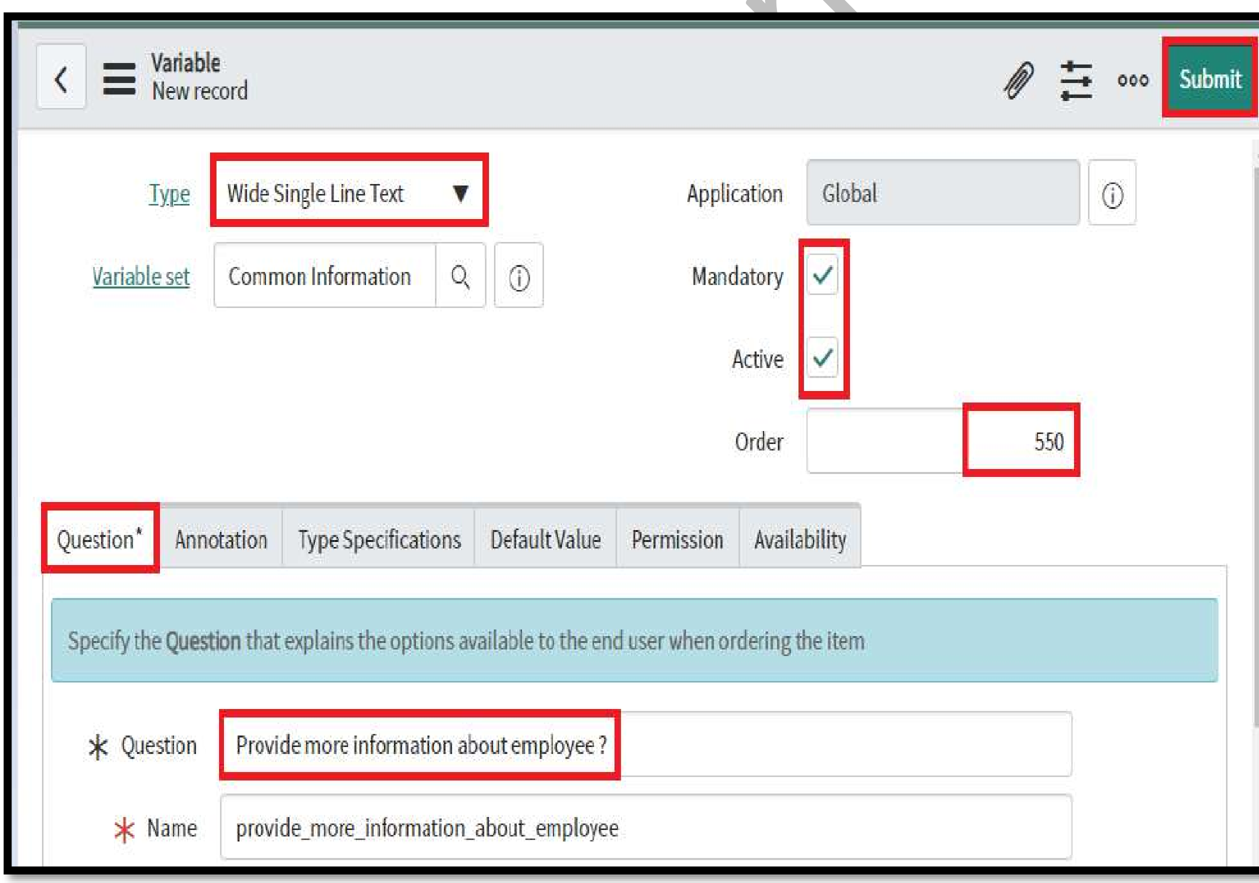
1. Navigate to **Service Catalog** > **Catalog Variables** > **Variable Sets > Common Information**



2. Open **Variable Tab** from **Related list**
3. Click on **New**



4. Fill **Wide Single Line Text** Variable



5. **Variable Type:** Wide Single Line Text
6. **Mandatory:** Checked **True**
7. **Order:** 550

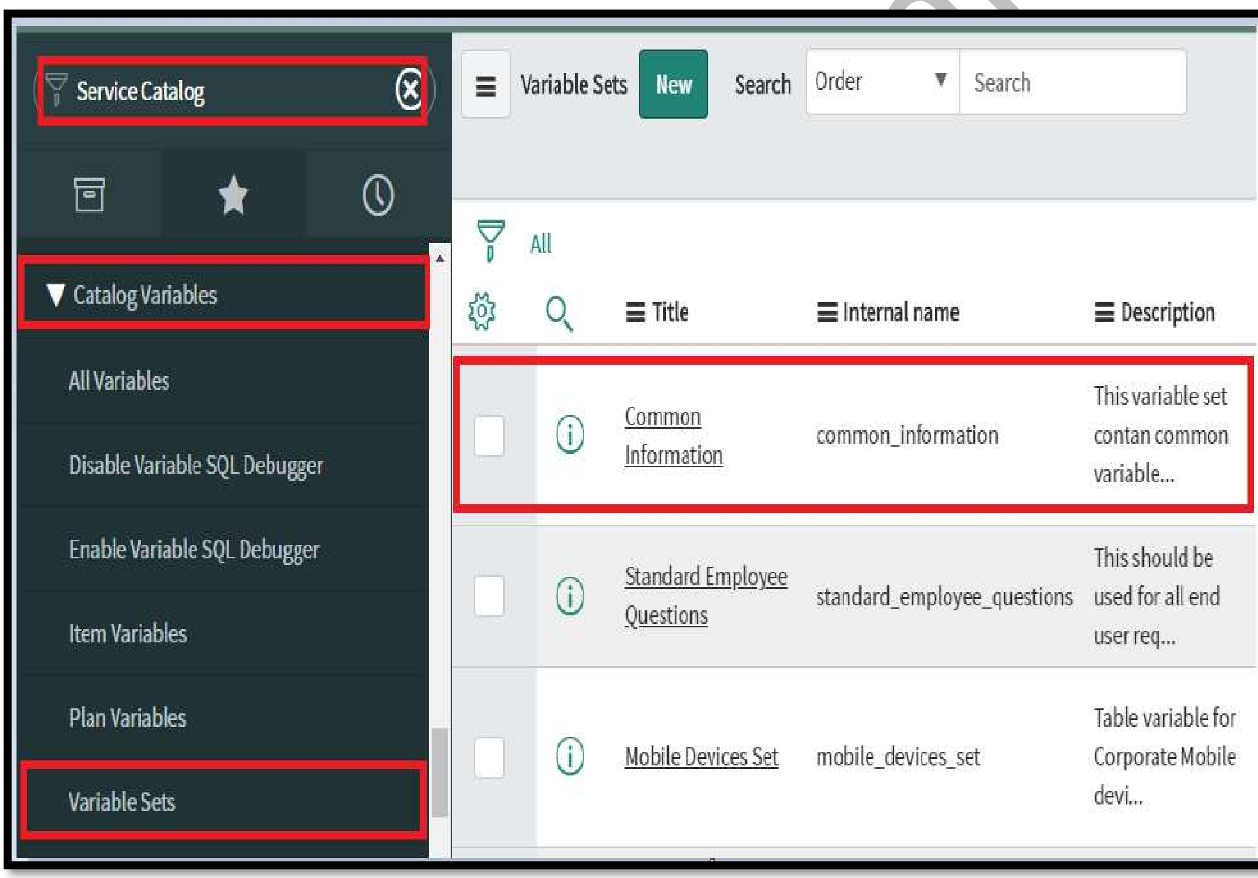
8. **Question:** Provide more information about employee?
9. **Name:** Auto Filled
10. Click on **Submit**

Working with IP Address Variable

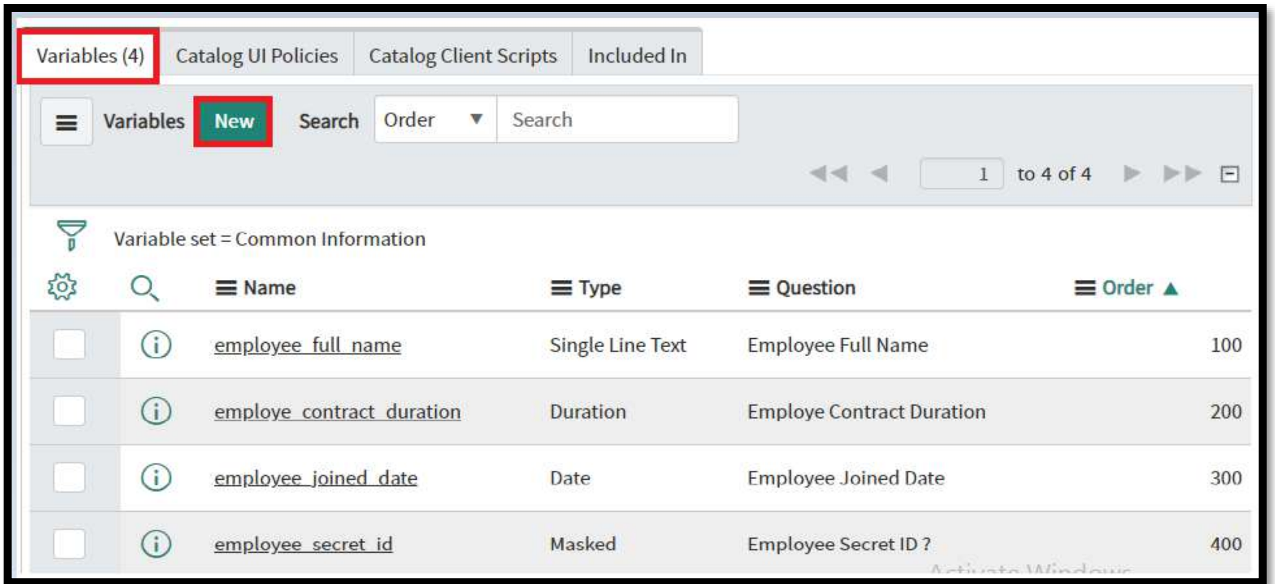
This variable is used to specify and get the **IPv4 and IPv6** data. Under the **Validation Scripts** submodule, a validation script also working on background.

Procedure

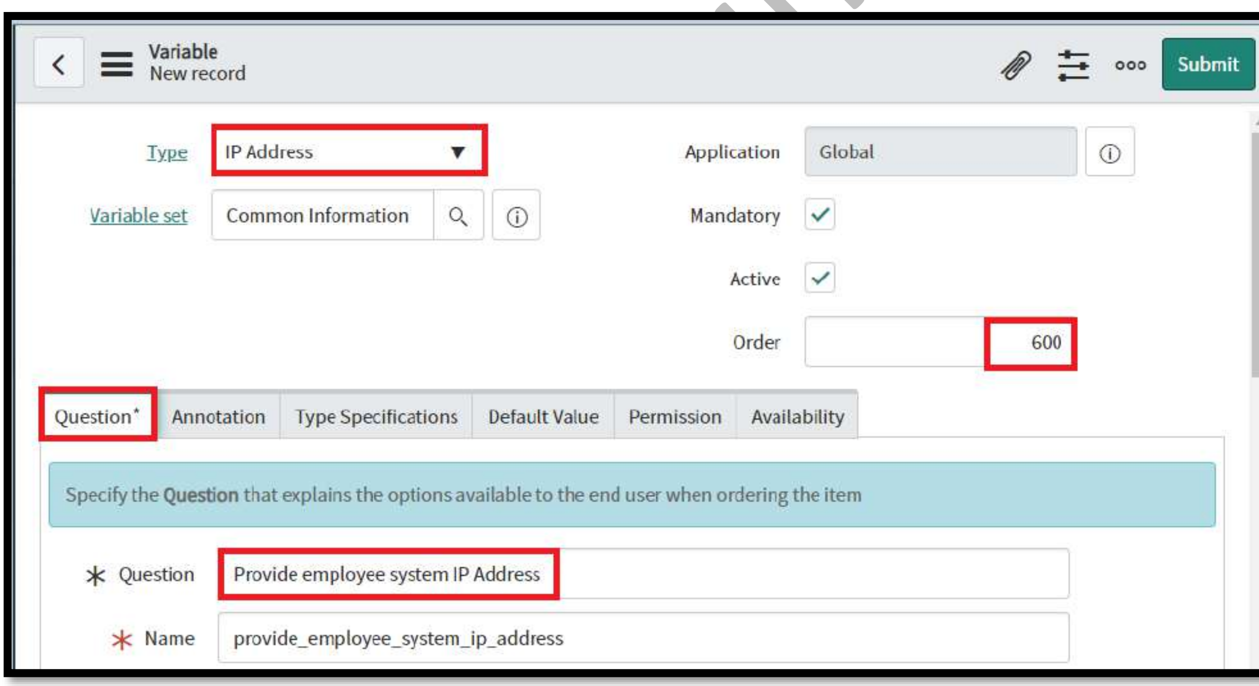
11. Navigate to **Service Catalog** > **Catalog Variables** > **Variable Sets > Common Information**



12. Open **Variable Tab** from **Related list**
13. Click on **New**



14. Fill **IP Address** Variable Form



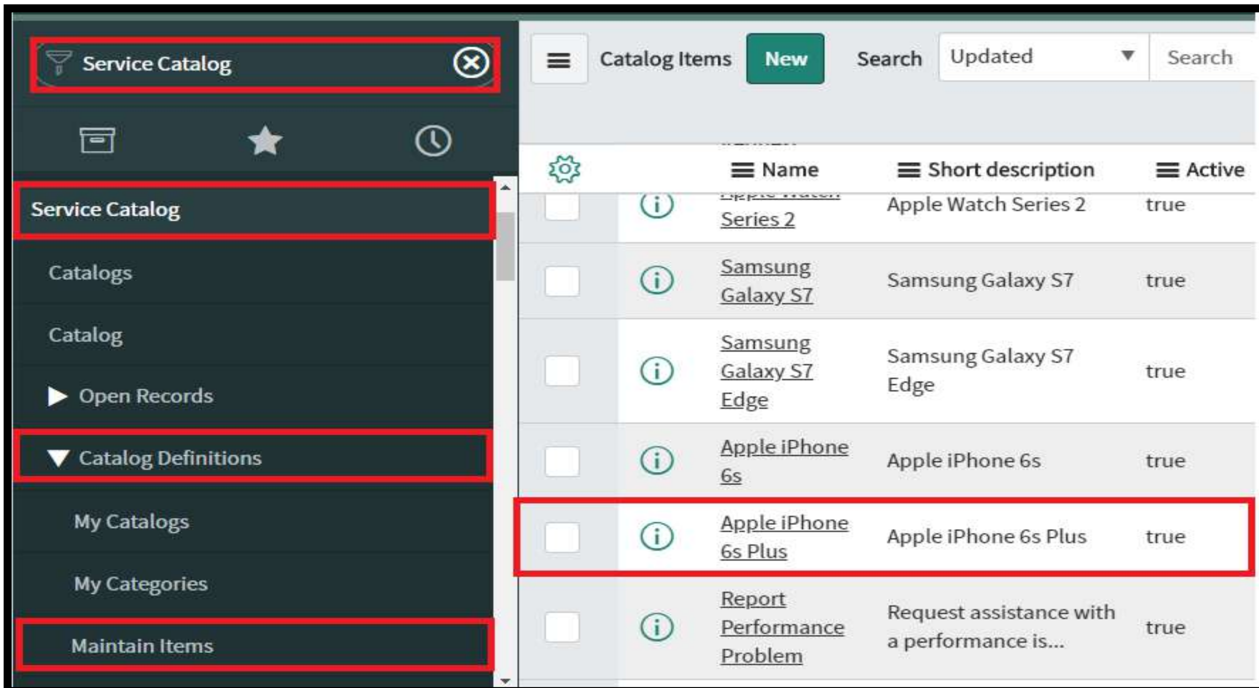
15. Variable Type: IP Address
16. Mandatory: Checked **True**
17. Order: 600
18. Question: Provide Employee IP Address?
19. Name: Auto Filled
20. Click on **Submit**

Exercise: 11 Add Variable Set to Catalog Item

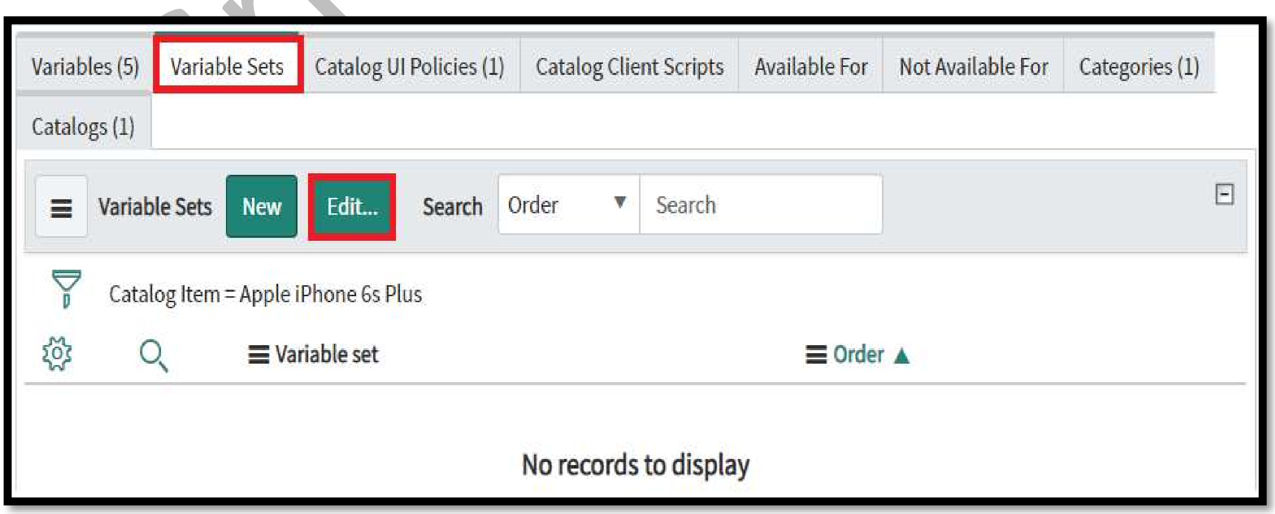
We are adding variable set to catalog Item for reusability purpose

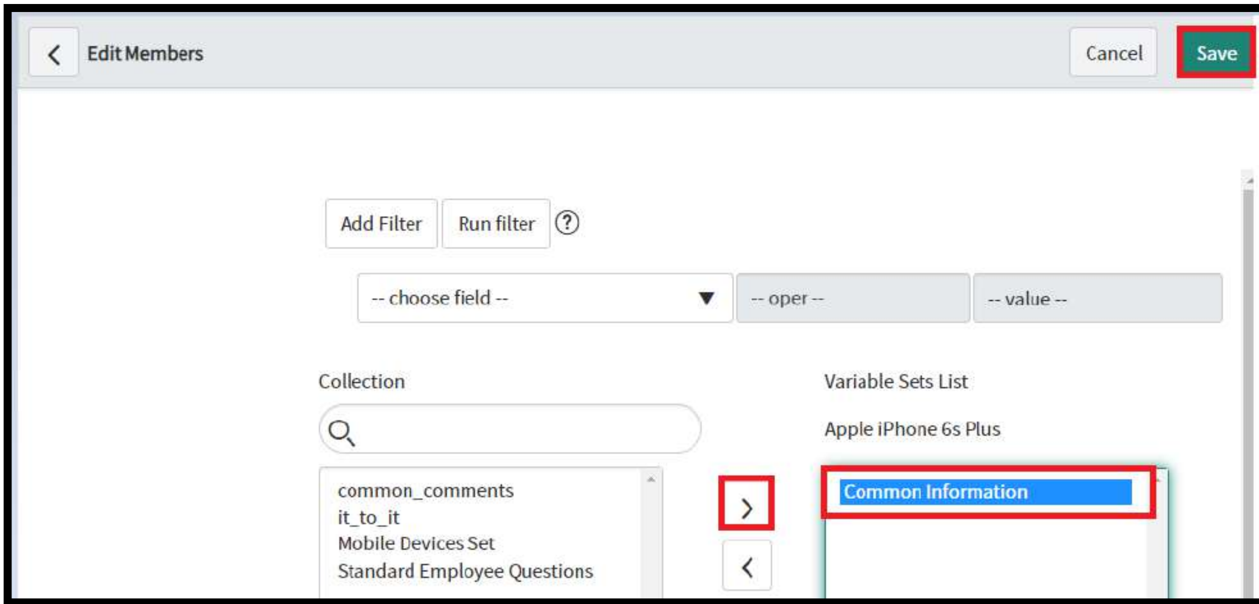
Procedure

1. Navigate to **Service Catalog** > **Catalog Definition** > **Maintain Items > Open I Phone 6s Plus item**

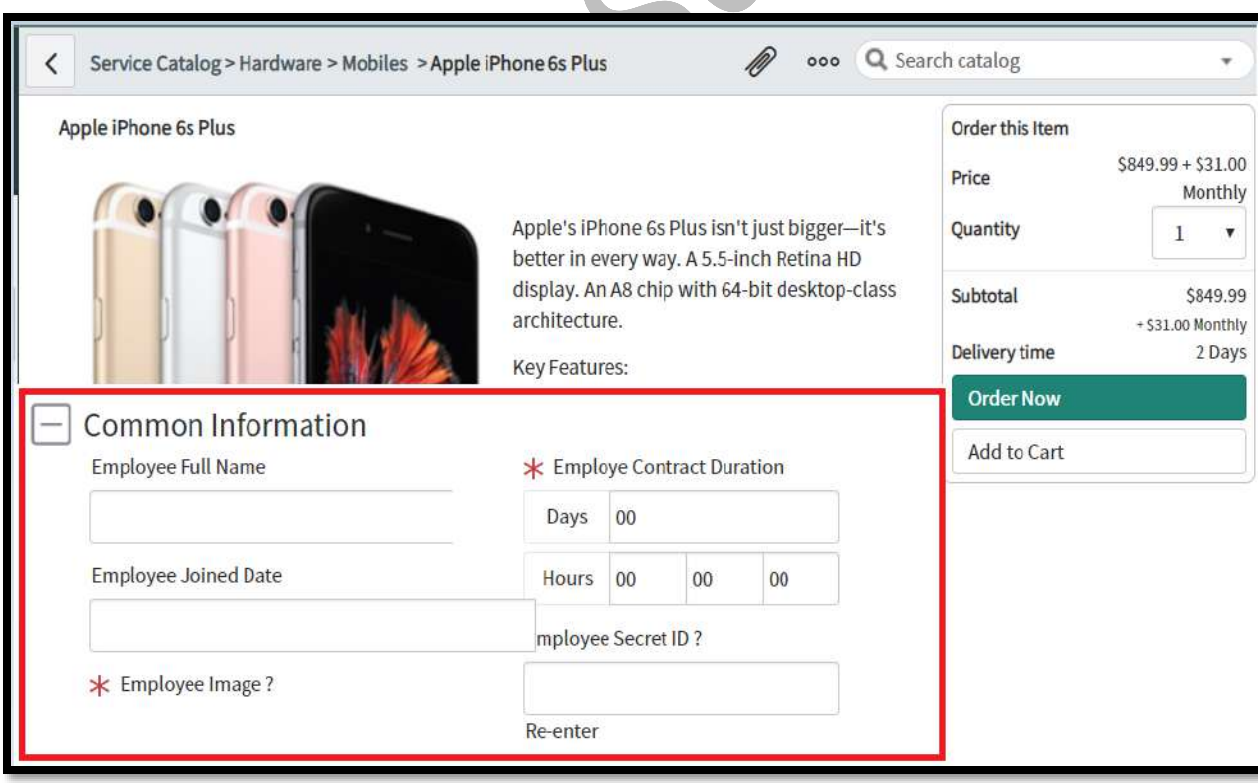


2. Open **Variable Sets Tab** from **Related list**
3. Click on **Edit**





4. Select **Common Information** variable sets from **Slash Bucket**
5. Click on **Right Arrow >**
6. Click on **Save**



7. Click on **Try It** and check results

Working with User Criteria

Service Catalog Application enables you to apply **User Criteria** to a service catalog item, which can be defining who **Can and Cannot** able access that particular item.

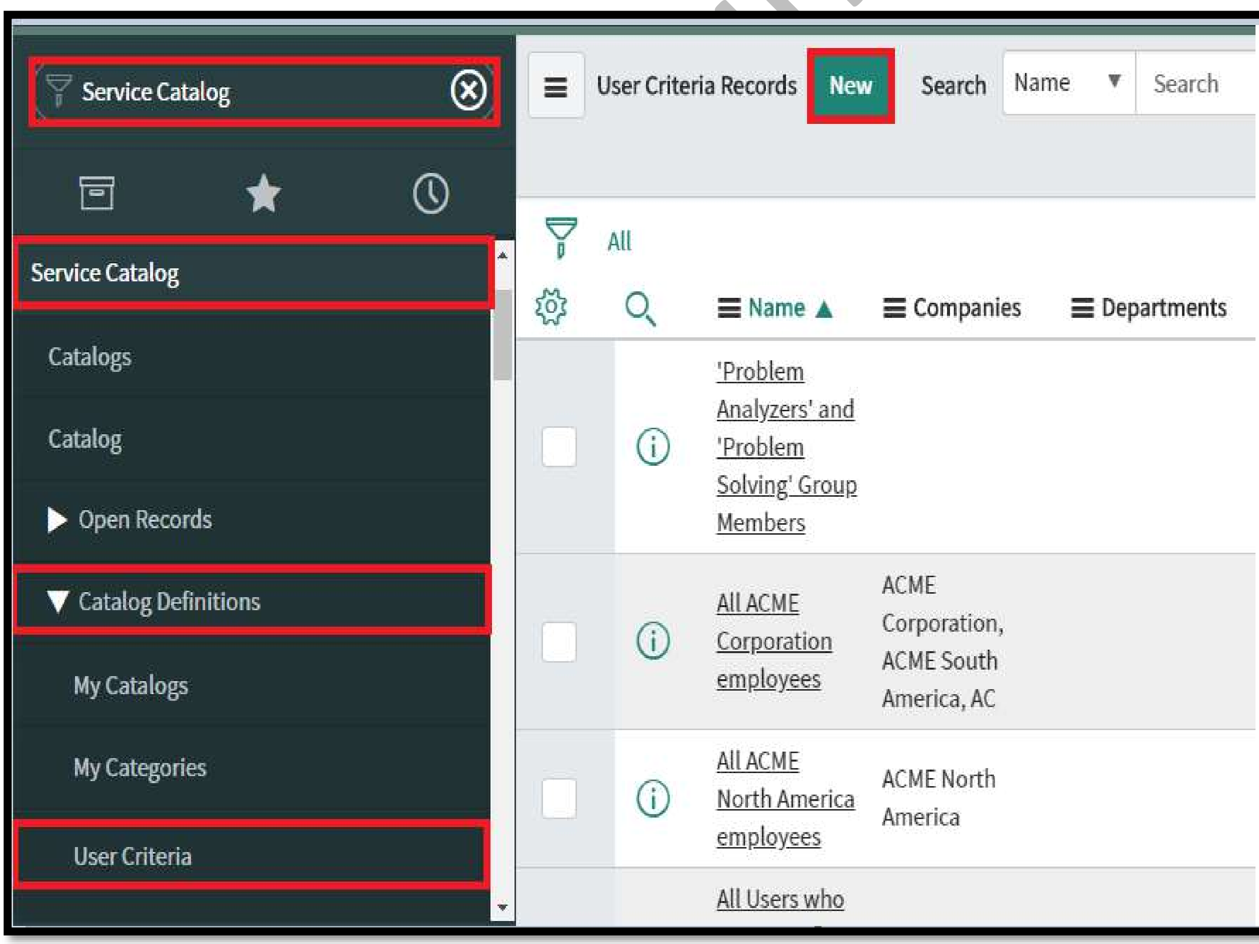
We are going to apply **User Criteria** to all types of catalog item, those are including **Record Producers, Order Guide, and Content Items.**

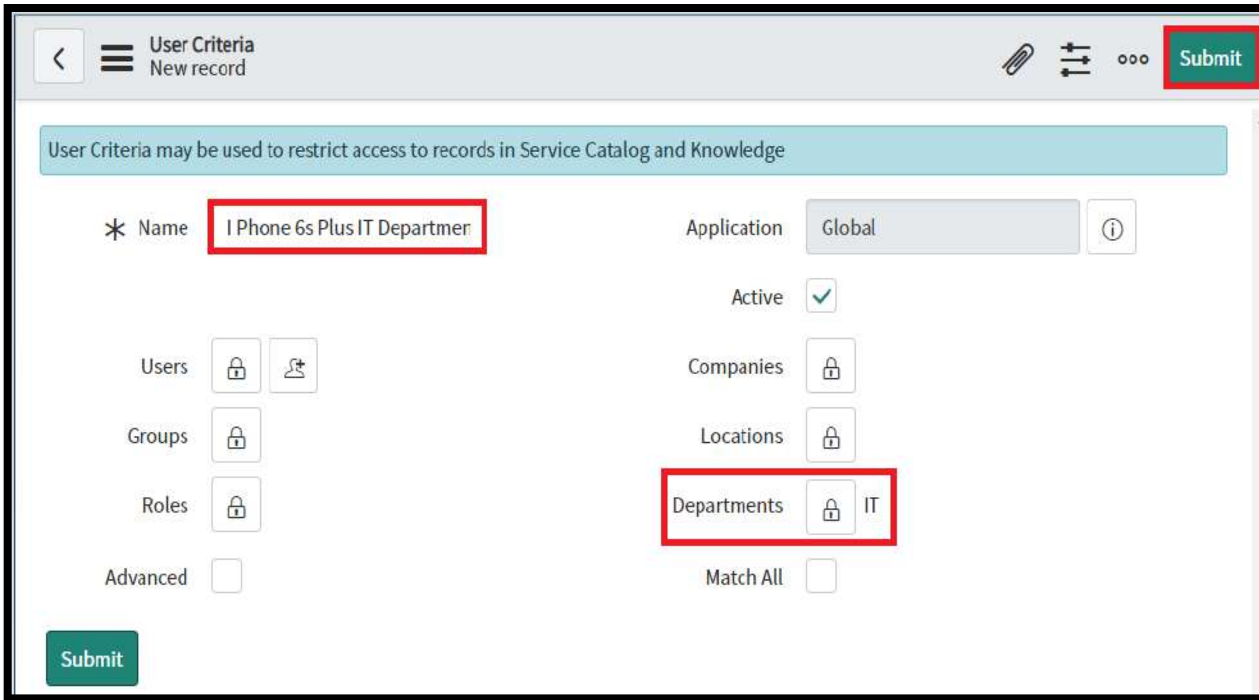
Exercise:12 Create new User Criteria

Task: I phone 6s Plus will be available for only **IT Department** people

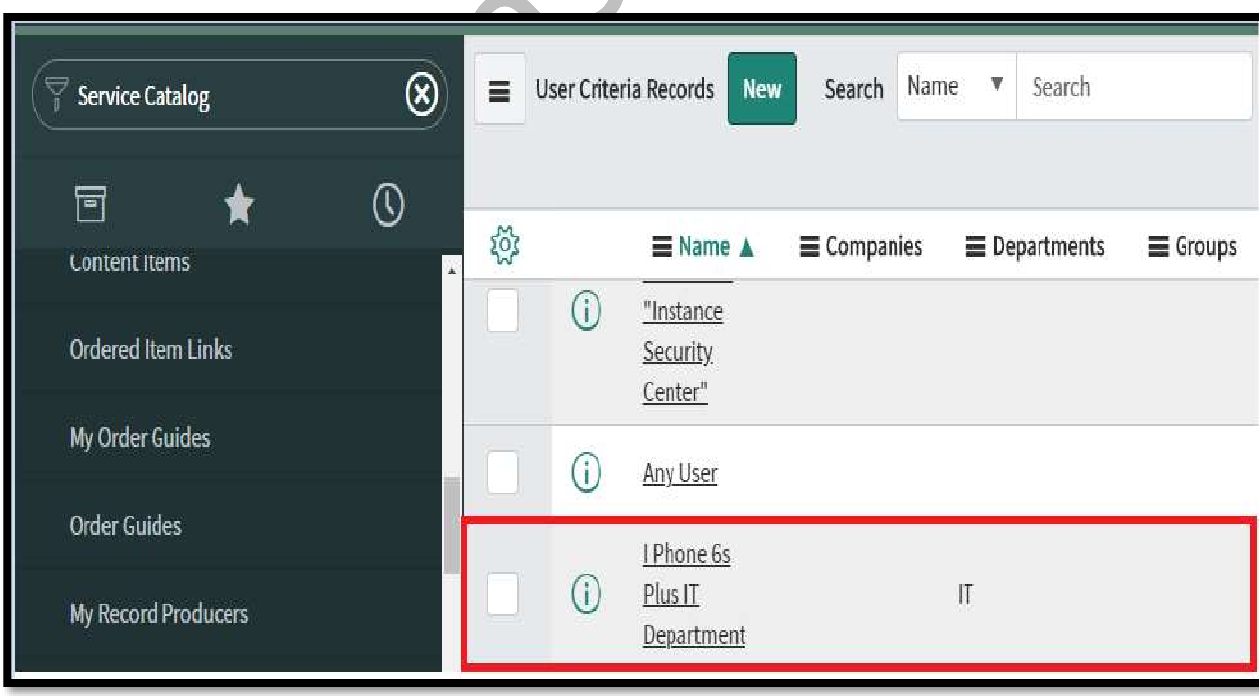
Procedure

1. Navigate to **Service Catalog** > **Catalog Definition** > **User Criteria**
2. Click on **New**





3. Fill **User Criteria** Form
4. **Name:** I phone 6s Plus IT Department
5. **Departments:** Select **IT**
6. Click on **Submit**



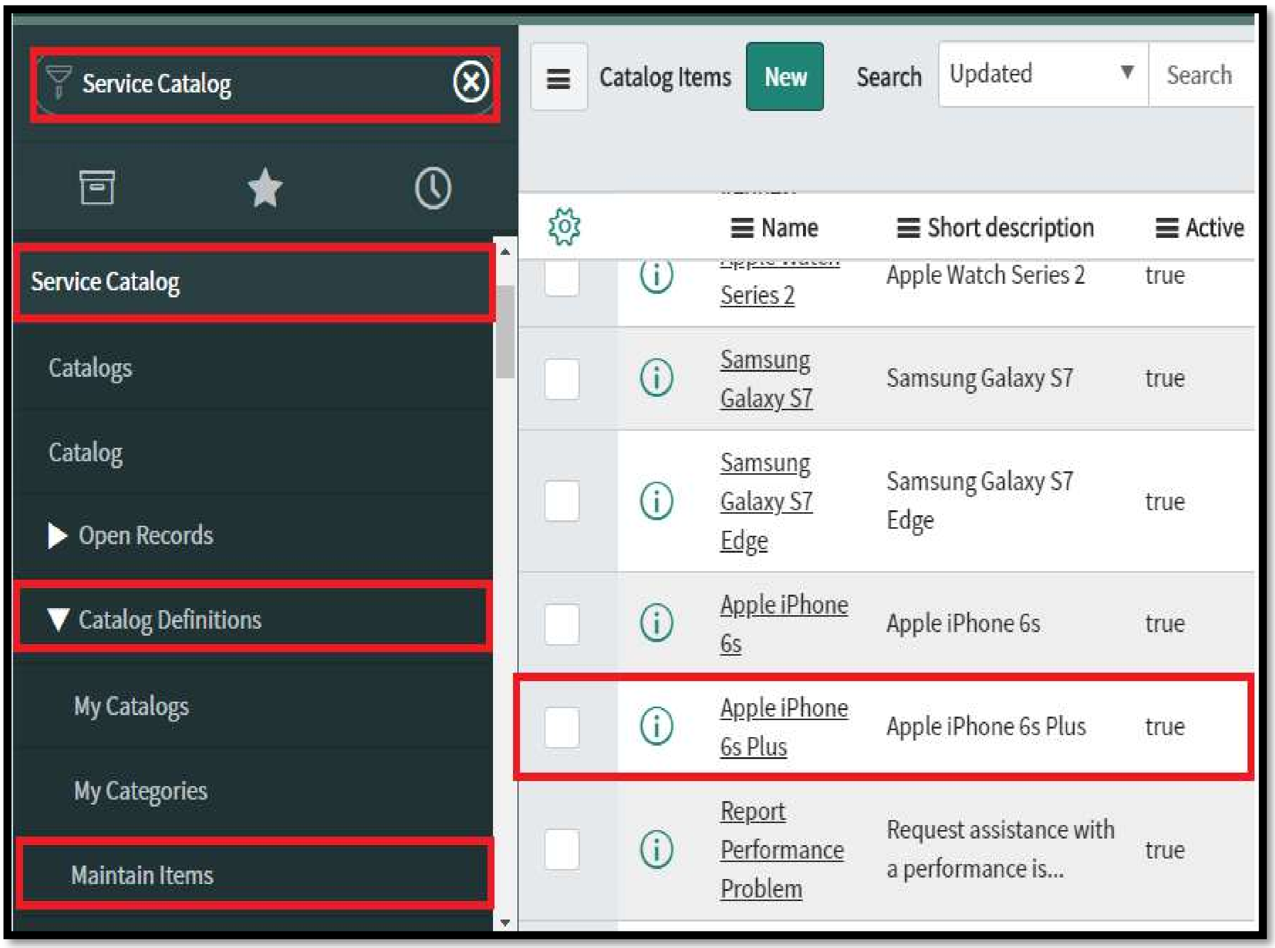
7. After added a **User Criteria** see look like above screen

Exercise: 13 Configure User Criteria for Service Catalog Item

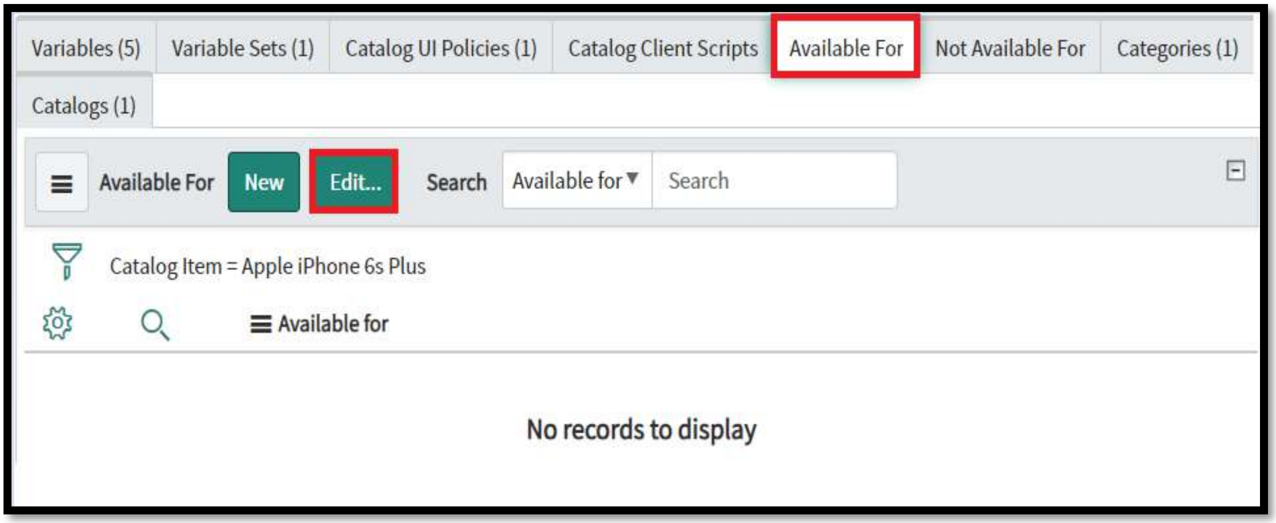
The user criteria used to restriction applies only within Service Catalog for the specific item it is applied for. However, the user criteria restriction is not applied outside the Service Catalog where the user has access to the item via the table.

Procedure

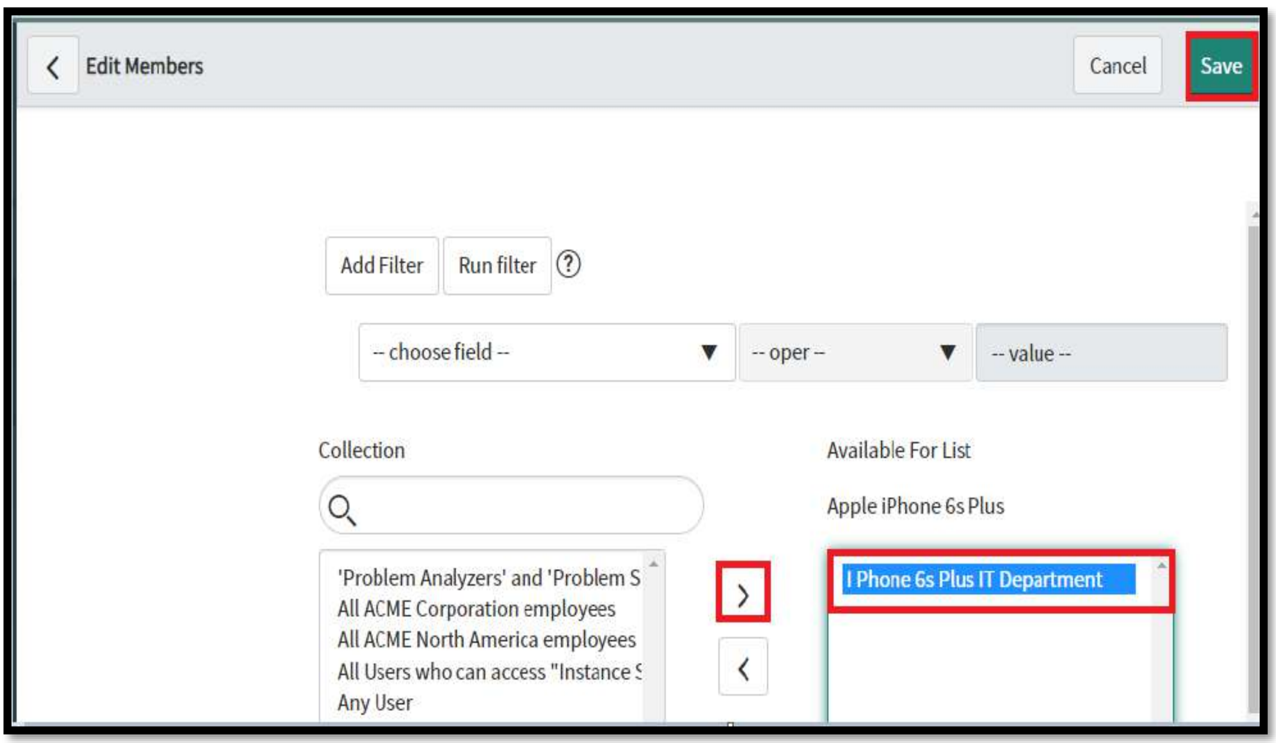
1. Navigate to **Service Catalog** > **Catalog Definition** > **Maintain Items > Open I Phone 6s Plus item**



2. Open **Available for** Tab from **Related list**



3. Click on **Edit**



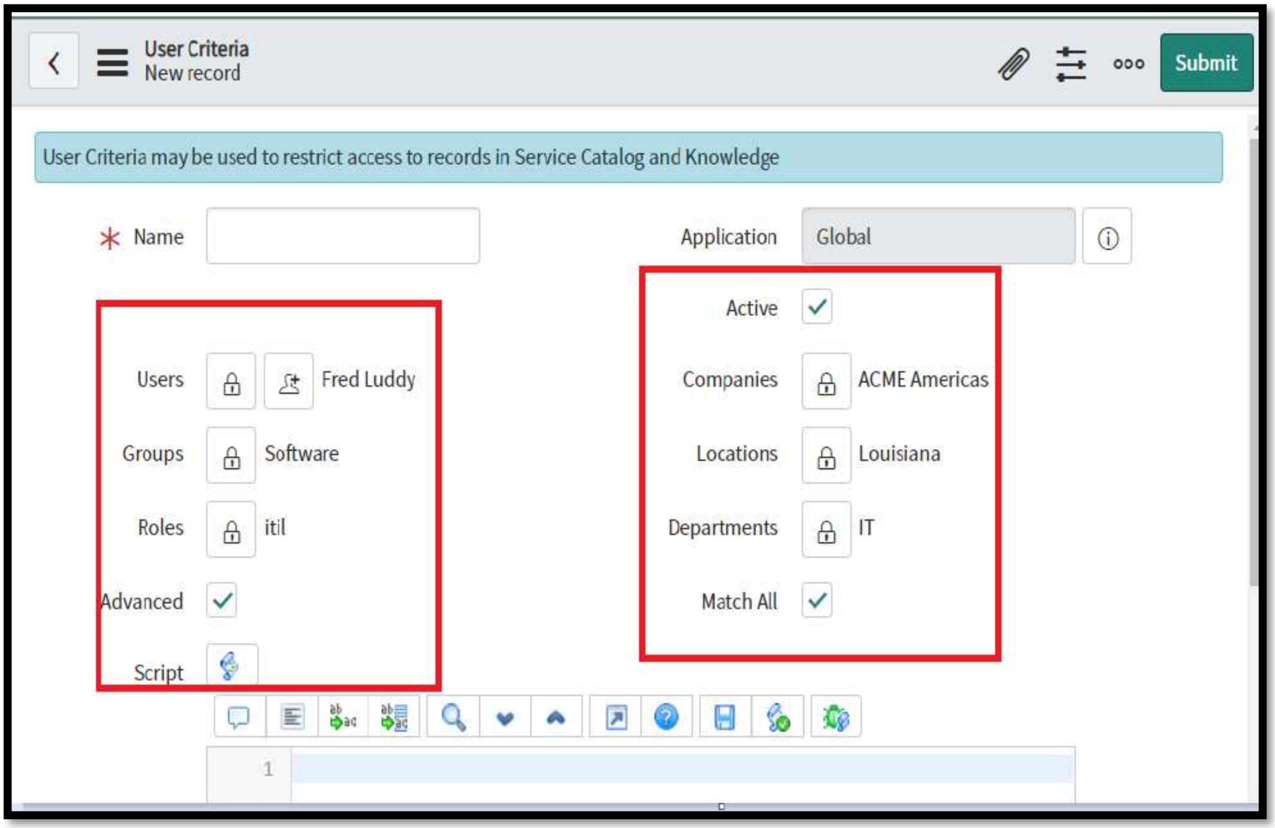
4. Select **I Phone 6s Plus IT Department** User Criteria from **Slash Bucket**
5. Click on **Right Arrow >**
6. Click on **Save**

Note: This catalog item should be visible to only **IT Department**

Important Note

The **Not Available for** property can override **Available for** Property. A user present on the **Not Available for** list then cannot access that item, even if that user is also on the **Available for** list for that item.

We can apply user criteria for particular category also, who can and cannot access category.



1. **Users:** Select users the **Catalog Item** should be visible to only particular users
2. **Groups:** Select Groups the **Catalog Item** should be visible to only particular group of peoples
3. **Roles:** Select Roles the Catalog Item should be visible to user have selected role
4. **Companies**: Select Companies the **Catalog Item** should be visible to only particular company peoples
5. **Location**: Select Locations the **Catalog Item** should be visible to only particular location peoples
6. **Departments**: Select Locations the **Catalog Item** should be visible to only particular Department peoples
7. **Advanced:** We can restrict users through script also

8. **Match All:** Match all above condition

Working with Service Catalog UI Policy

Service Catalog UI Policy is similar to UI policy controlling fields and changing form behavior and dynamically apply to variable validation. Catalog UI policies can be applied to a **Catalog Item or a Variable Set.**

We can perform below operations using by Service Catalog UI Policy

1. Mandatory/Option
2. Readable/Writable
3. Visible/In Visible

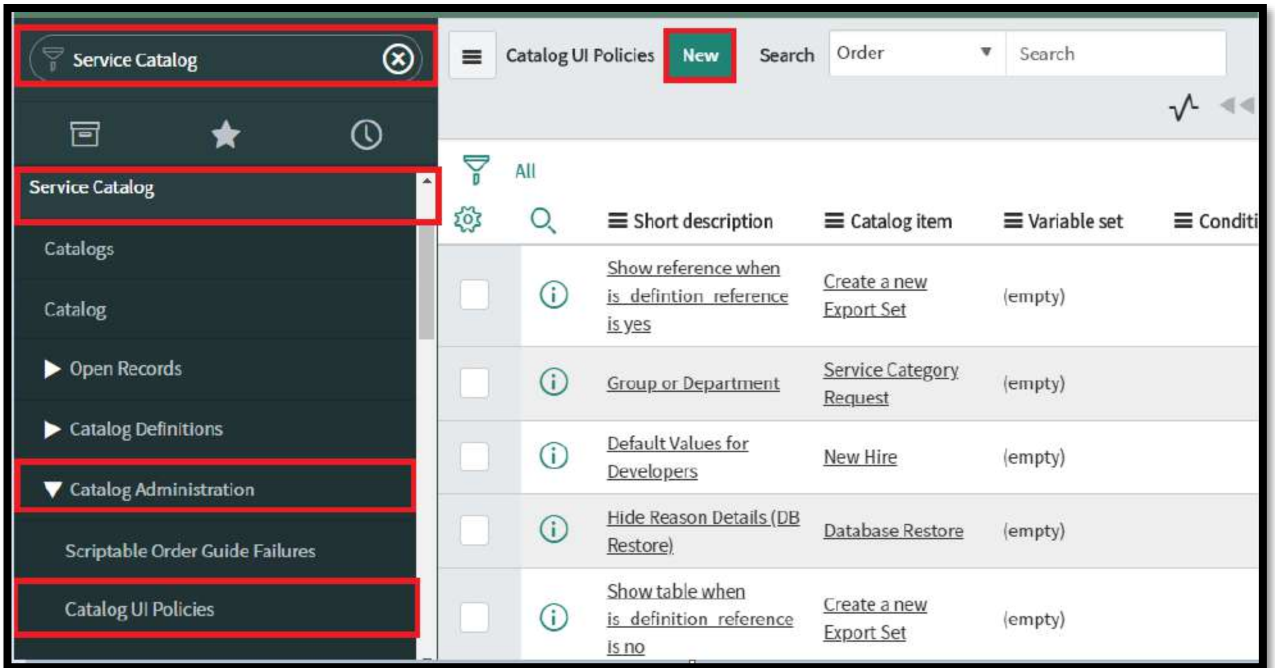
Exercise: 14 Configure a Catalog UI Policy for Catalog Item

We can apply **Catalog UI Policies** for catalog items when any customer ordered in the service catalog then change form behavior based on business requirement

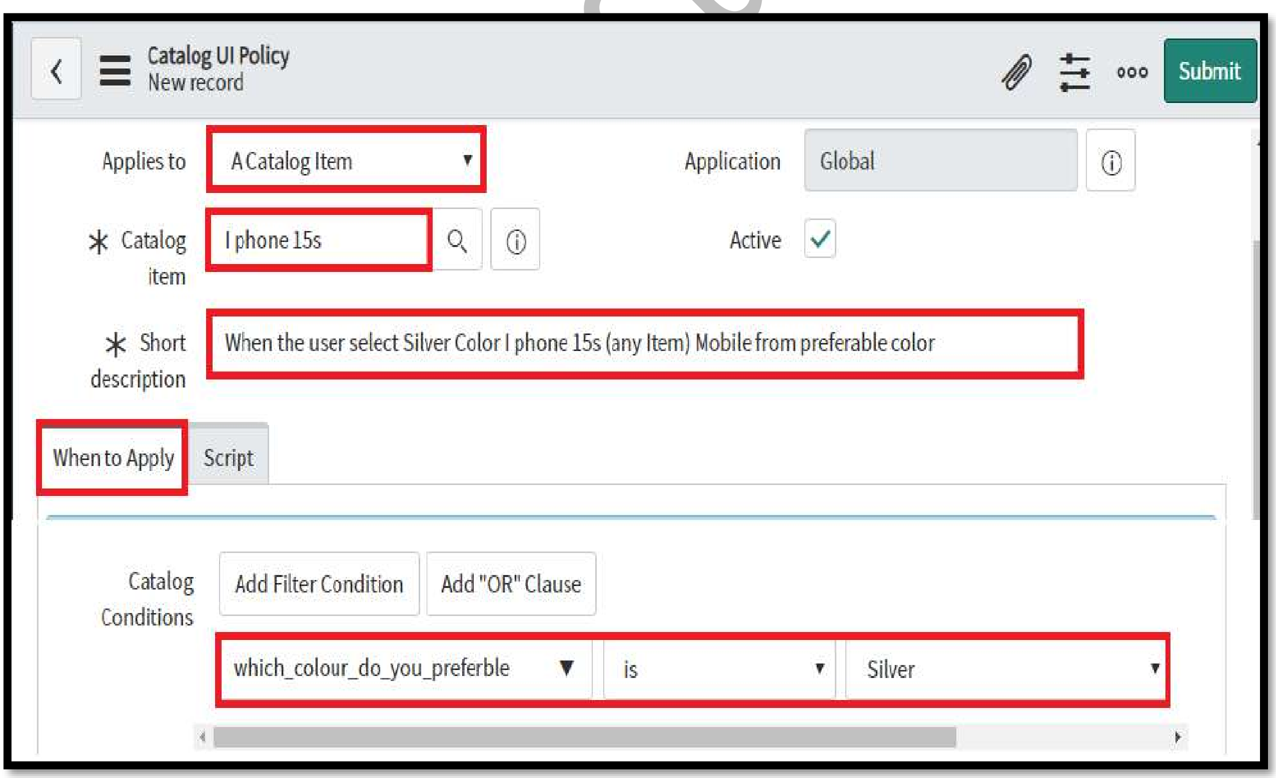
Condition: When the user select **Silver Color I phone 15s (any Item)** Mobile from preferable color
Action: Extended Memory up to variable should not be visible

Procedure

1. Navigate to **Service Catalog > Catalog Administration > Catalog UI Policies**
2. Click on **New**

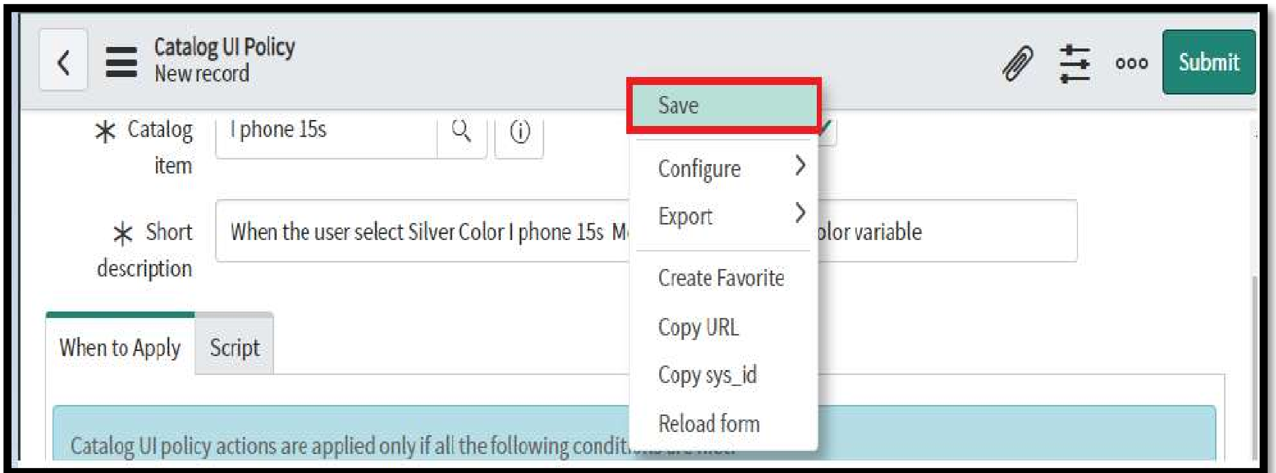


3. Fill **Catalog UI Policy** form

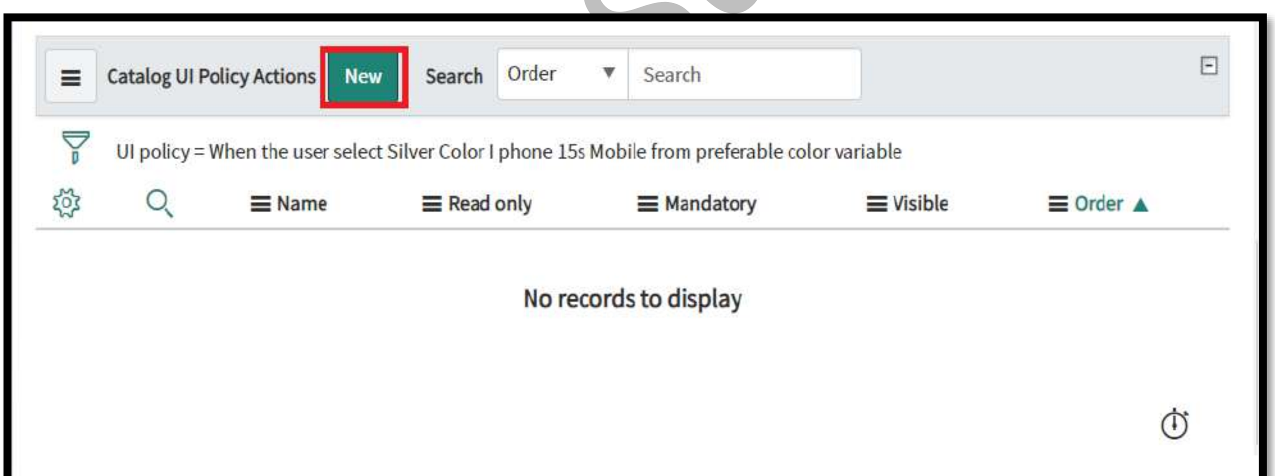


4. **Applies to:** A Catalog Item
5. **Catalog Item:** I Phone 15s

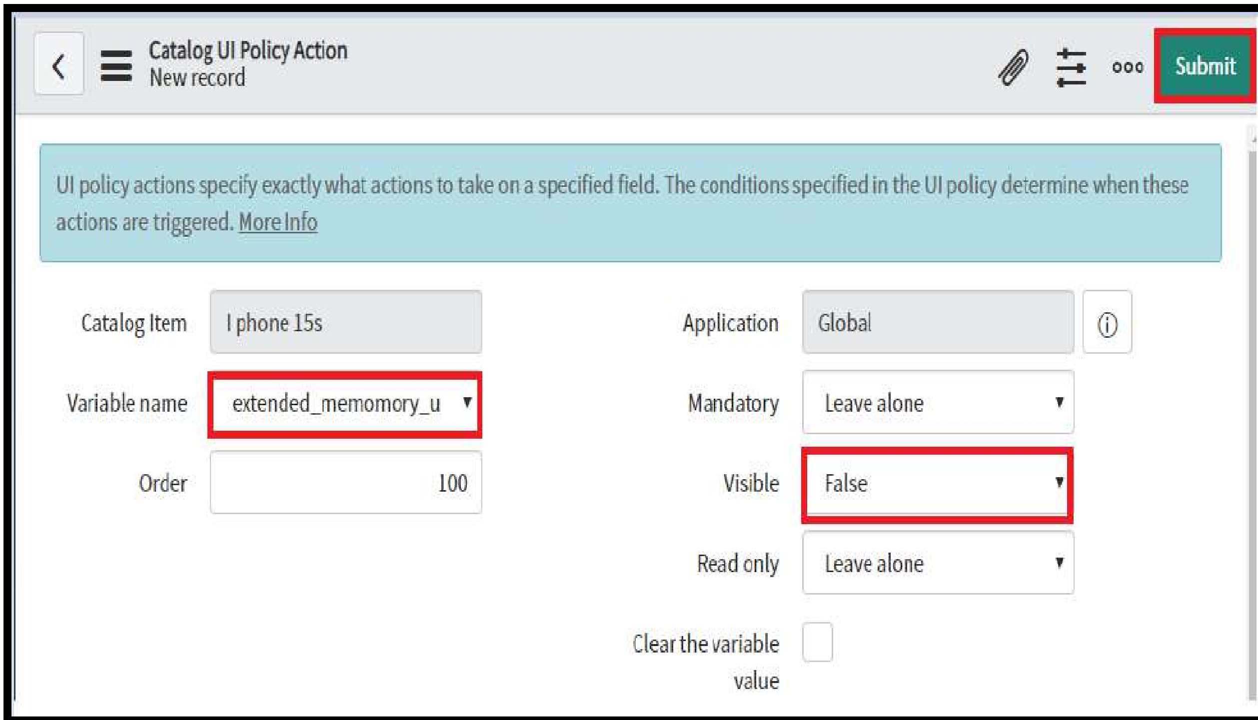
6. **Short Description:** When the user select Silver Color I phone 15s Mobile from preferable color variable
7. **When to Apply:** Condition [Which color do you preferable] **[is] [Silver]**
8. **Right Click** on form header and click on **Save**



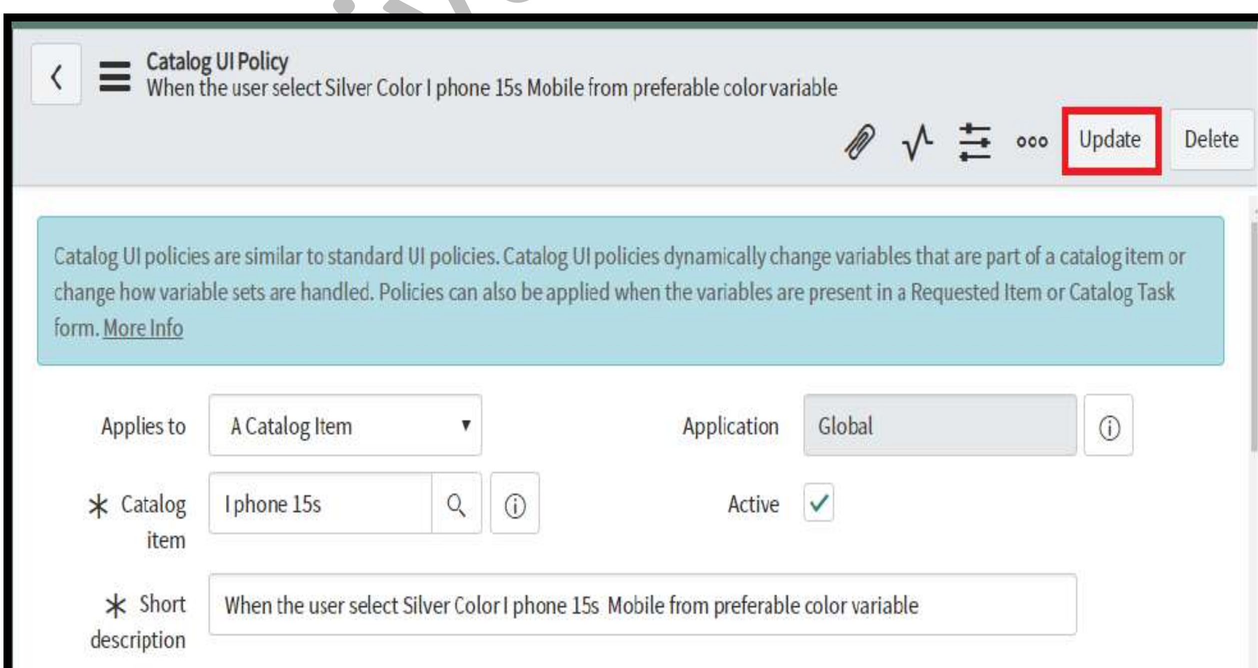
9. **Catalog UI Policy Action** Section



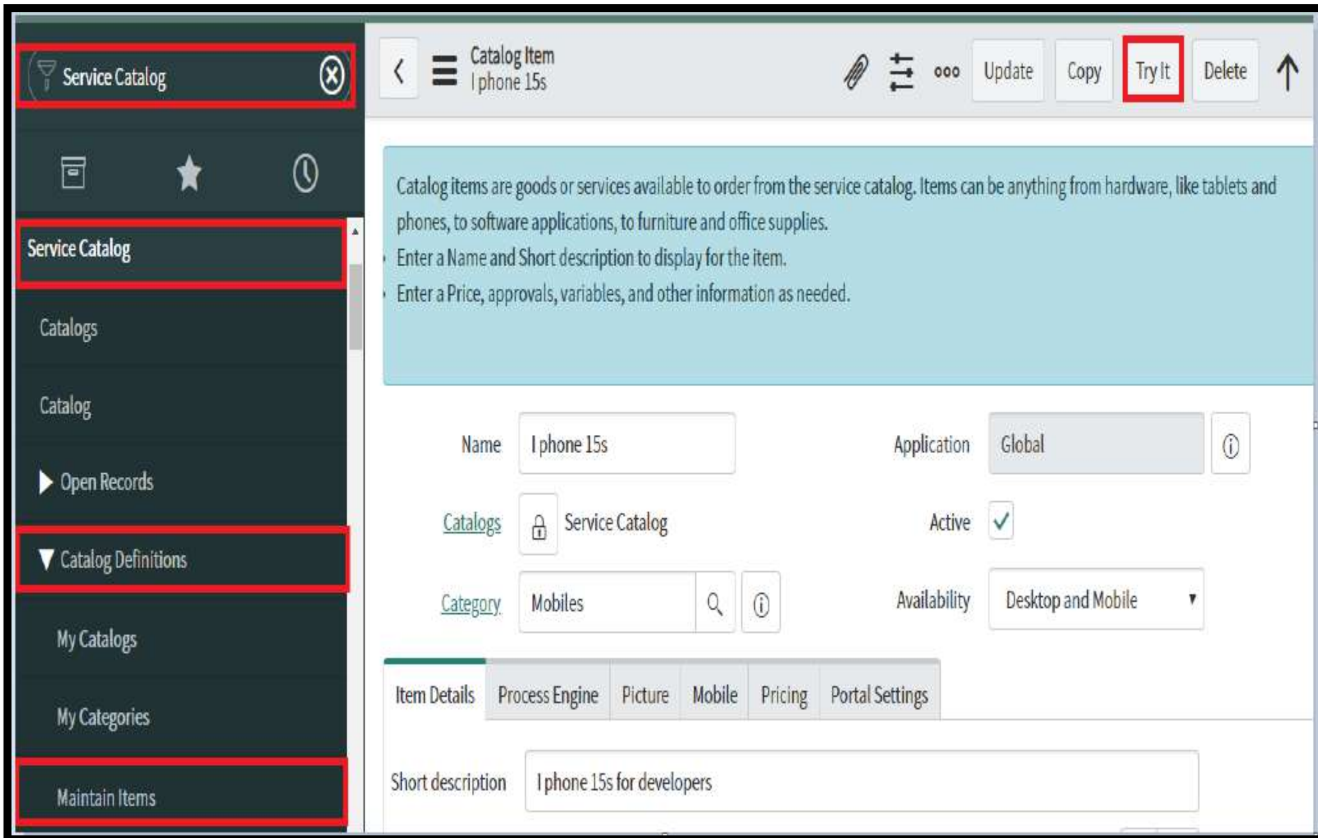
10. Click on **New**



11. Variable Name: Select variable name which one needs to be hide
12. Visible: False
13. Order: 100
14. Click on **Submit**
15. Click on **Update**



16. Navigate to **Service Catalog > Catalog Definition > Maintain Item > I Phone 15s Item**
17. Click on **Try It**



18. Select preferable color value is **Silver**

19. Expected variable is **Hidden** like above screen shot

Chapter – 17

Workflows

What is Workflow

Service now Workflow will provide a **drag-and-drop** interface for automating multi-step processes across the platform.

Each workflow will contain number of a sequence of activities, such as generating records, notifying users of pending approvals, or running scripts.

Life cycle of Workflow

A workflow will run when a triggering event occurs. The common triggers include a record being inserted or updated into a specific table, or a particular field in a table being set to a specified value.

When a workflow activity completes, then the workflow transitions to next activity. An activity might have several different possible transitions to various activities, depending on the outcome of the activity.

For example, already you might create a workflow that runs whenever a user requests approval for an item they want to order from the catalog. We can also schedule workflows to run periodically or call them from scripts such as BR rules.

The graphical Workflow Editor can represent workflows visually as a type of flowchart visualization manner. It will show activities as boxes labelled with information about that activity and transitions from one activity to the next as lines connecting the boxes.

Who can create and modified Workflows

The person should have below roles to **create, modify, delete, and publish workflows.**

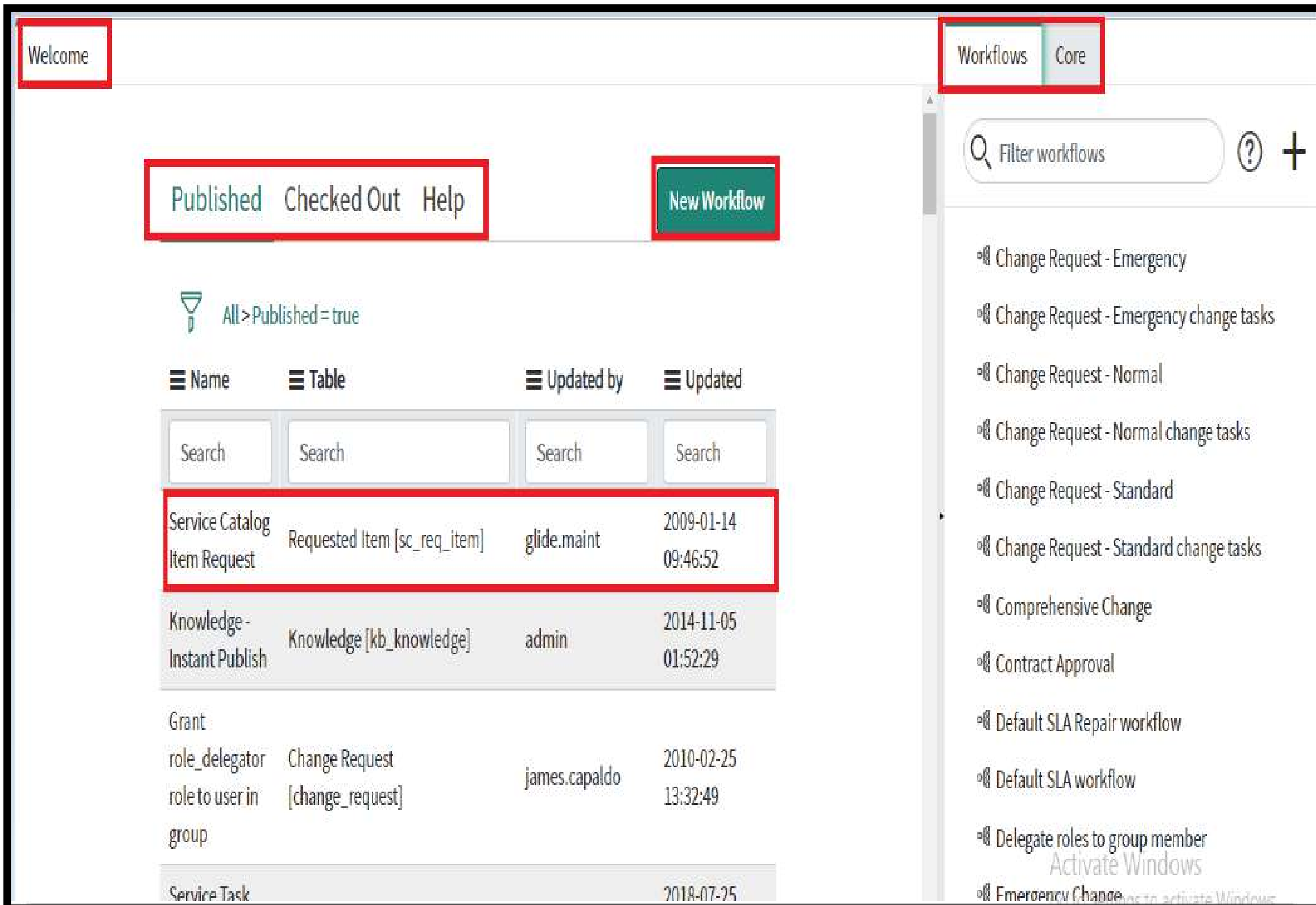
1. **workflow_creator**
2. **workflow_admin**

Workflow editor

The Workflow Editor is an interface for creating new workflow and modifying existing workflows by arranging and connecting activities to drive processes.

Workflow Home Screen

1. Servcenow admin and Developers can working with workflow editor
2. Service now admin can open with the Welcome page, which can be displays a list of active, and How it would be published.
3. We can open existing workflows, create new workflows, and open help resources related to workflow.



Published: Click to view list of published workflows in our instance

Checked Out: Click to view list of workflows checked out to current logged in user

Help: Click to view links to help resources for workflow to implementing new

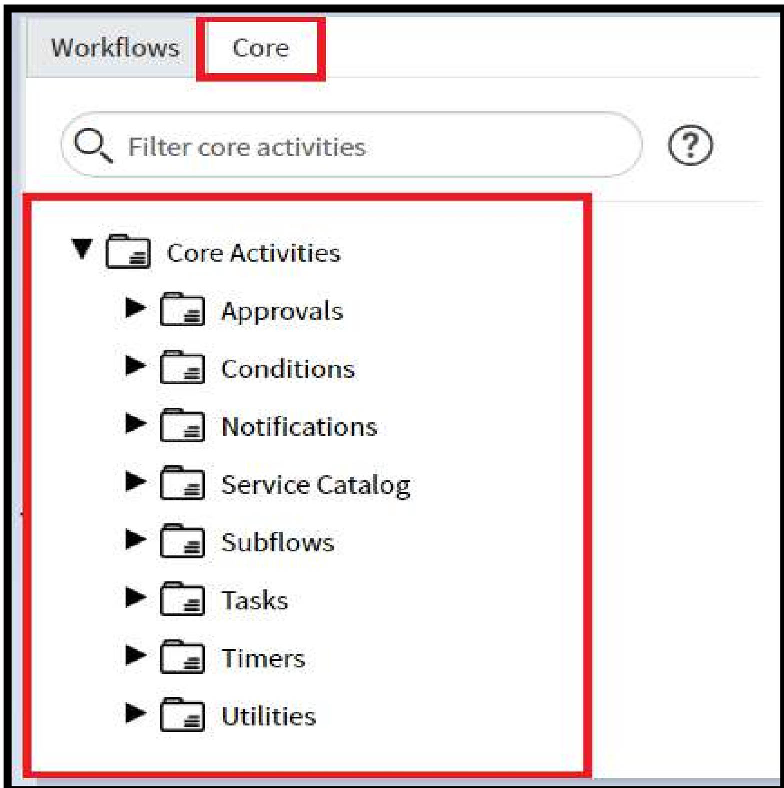
New Workflow: Click to create a new workflow

Core: Available workflow activities appropriate for the selected workflow. Contents can include activities provided by the base system and those purchased with orchestration.

Core Activities in Workflow

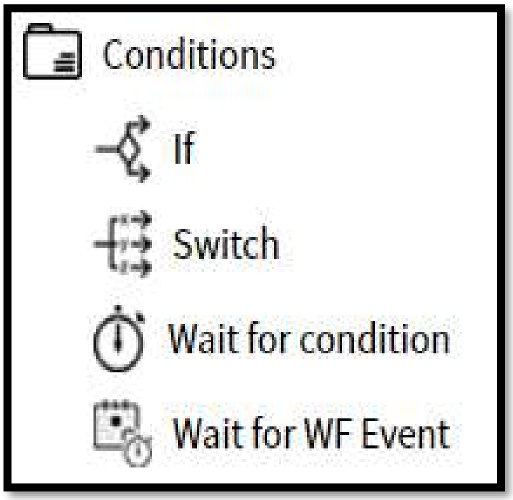
The Core activity tab will contain the standard activities available by default to all workflows and any activities purchased with Orchestration, organized by category.

Click the arrow icons to expand or collapse the activity lists under each category. To add an activity to a workflow, drag it to the canvas. For more information, See below

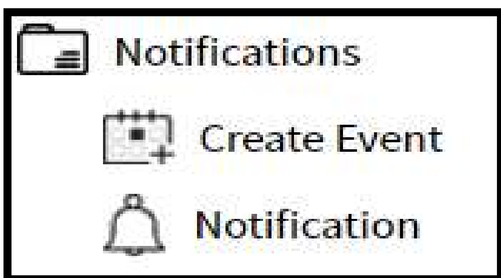


1. Approval Activity

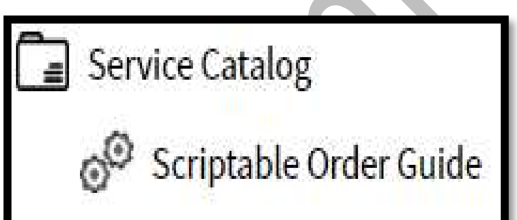
2. Condition Activity



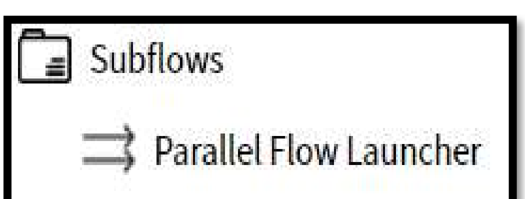
3. Notification Activity



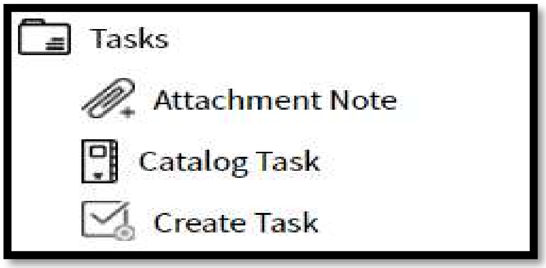
4. Service Catalog Activity



5. Sub flow Activity

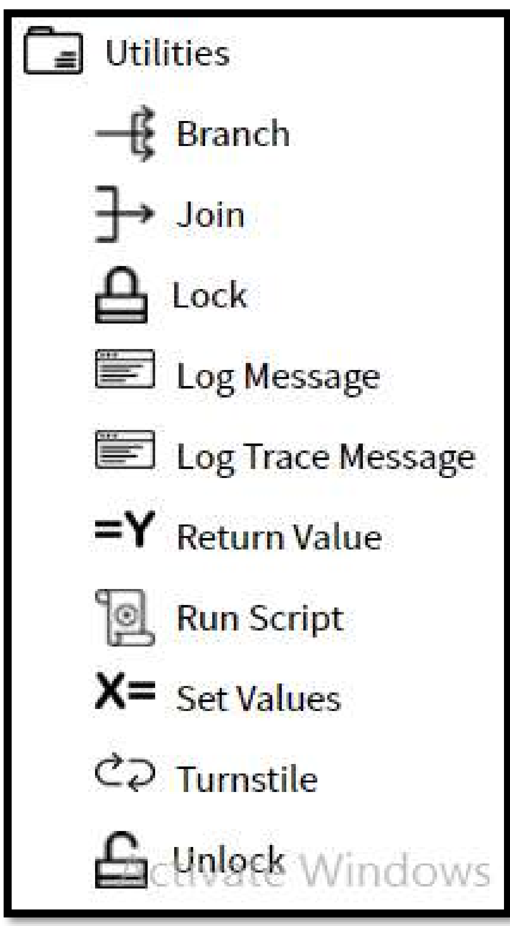


6. Tasks Activity



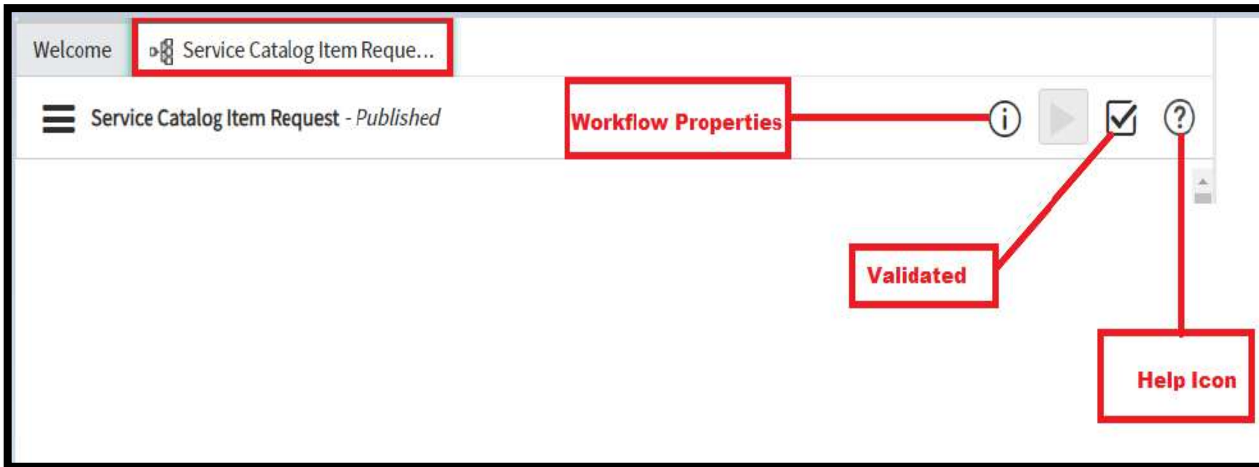
7. Timer Activity

8. Utilities Activity



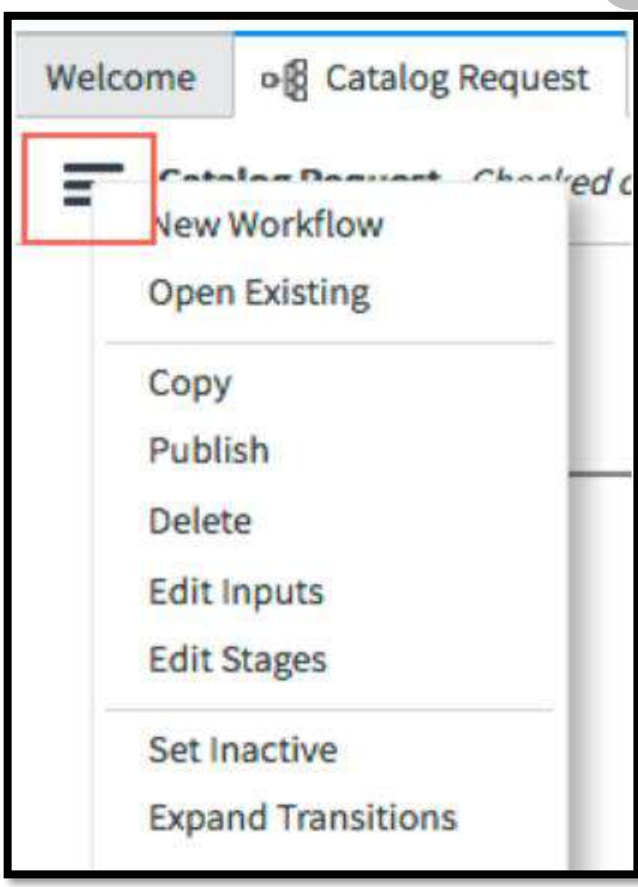
Workflow editor title bar options

When we working with workflow using these all tabs frequently



Workflow Context Menu

Workflow context menu can provide additional features to configure the workflow.



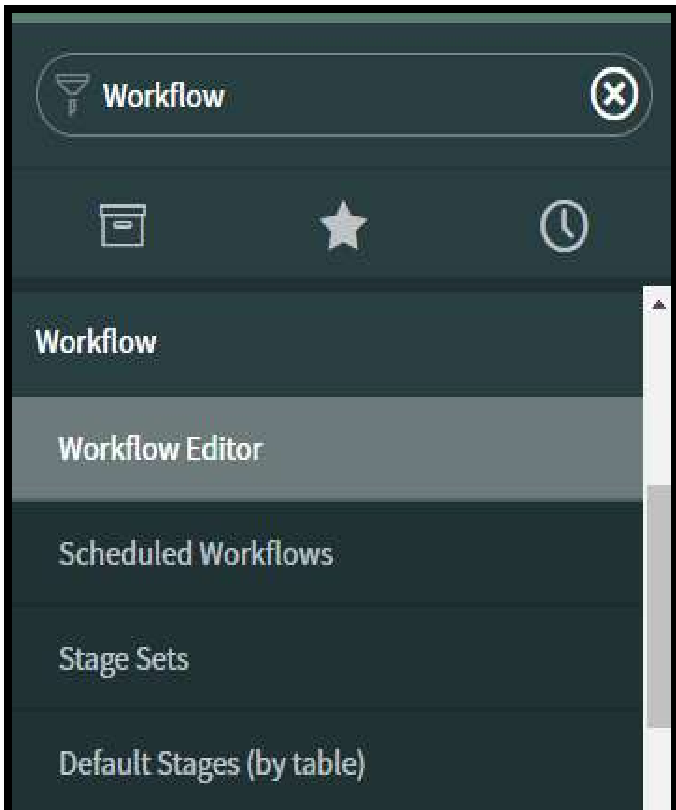
Exercise: 1 Creating a new workflow

We can create new workflow that automate a multi-step process from **Workflow Editor.**

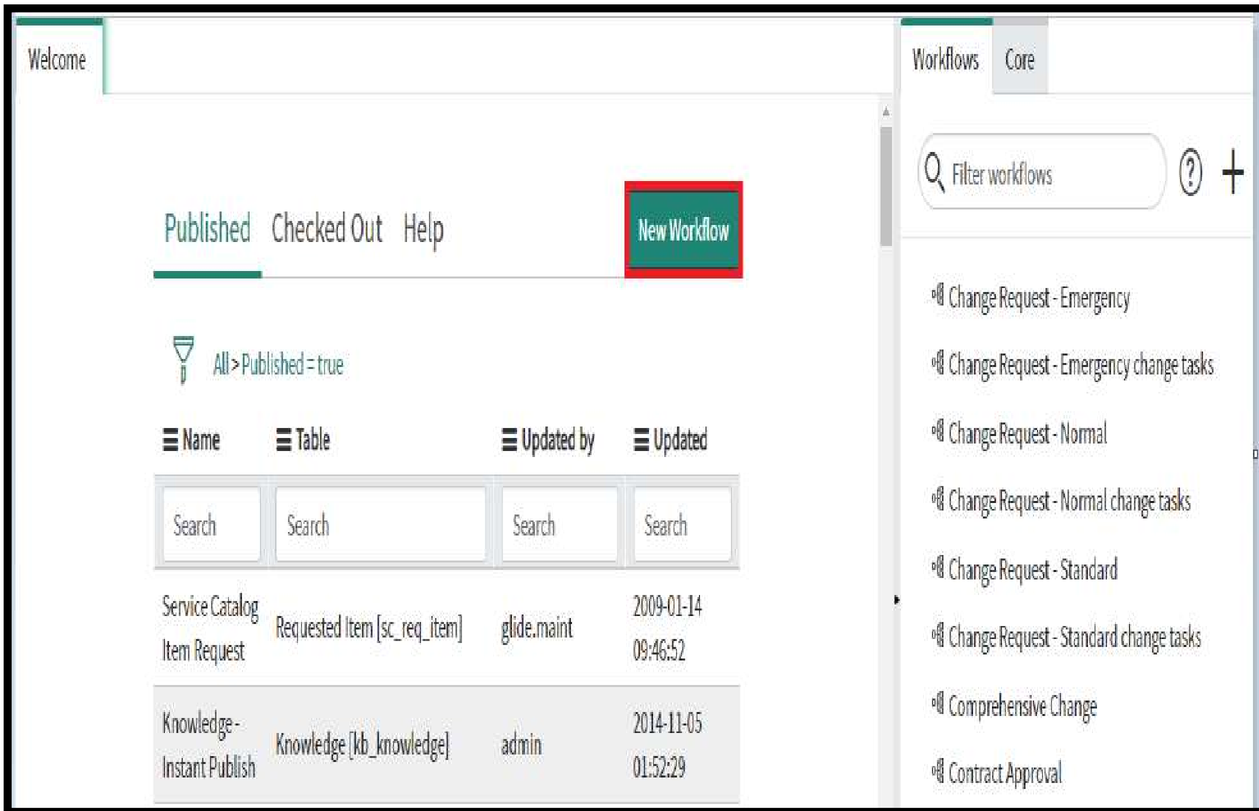
After created this workflow then we can add it to our **I Phone 12s Catalog Item**

Procedure

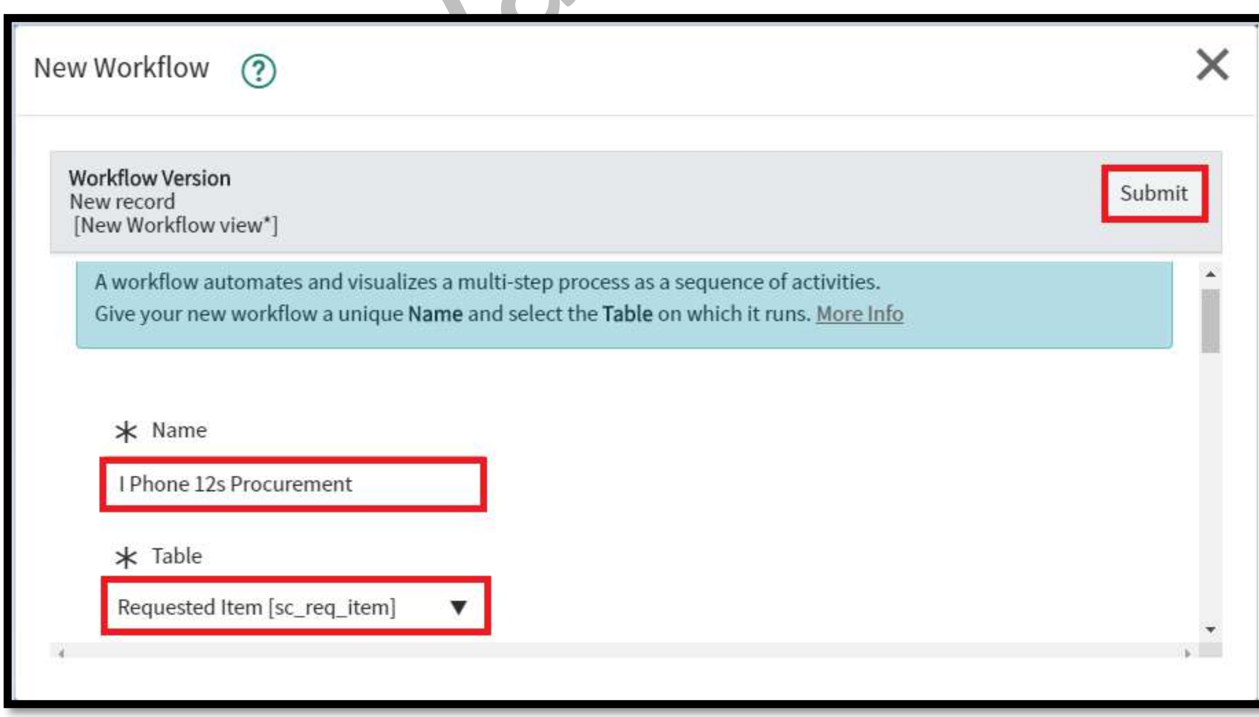
1. Navigate to Workflow > Workflow Editor
2. Click New Workflow

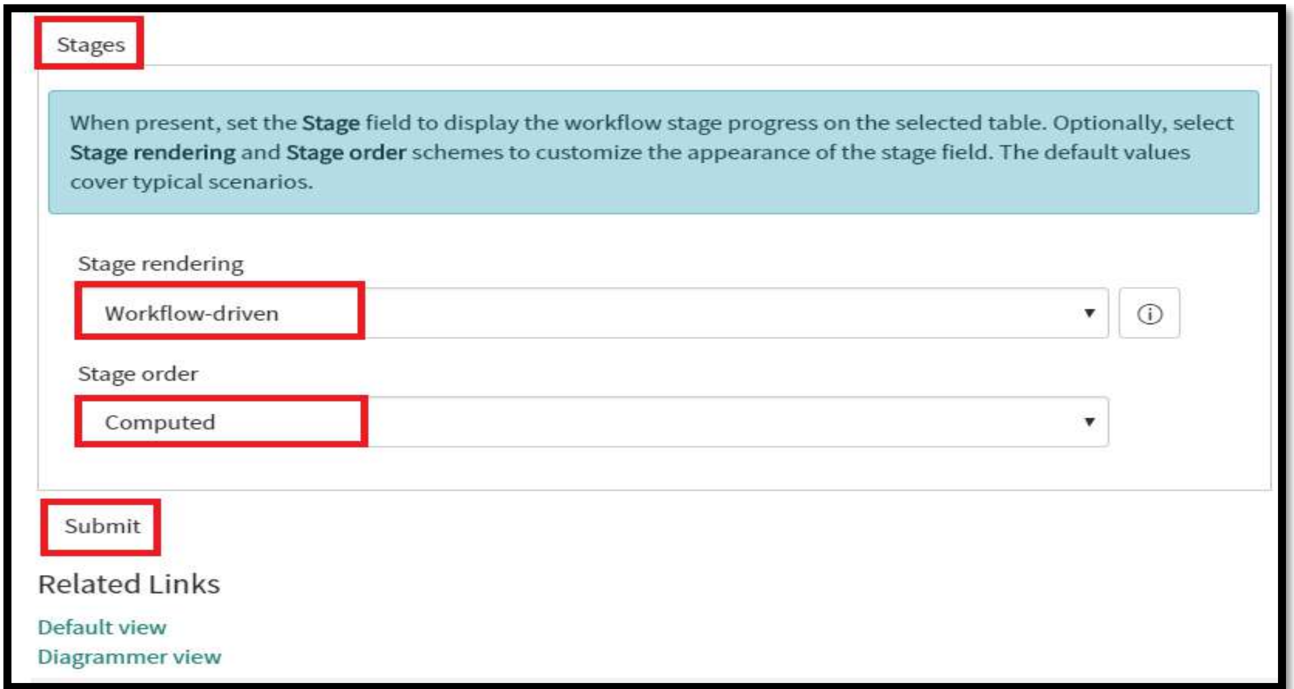


3. Click on **New Workflow** button

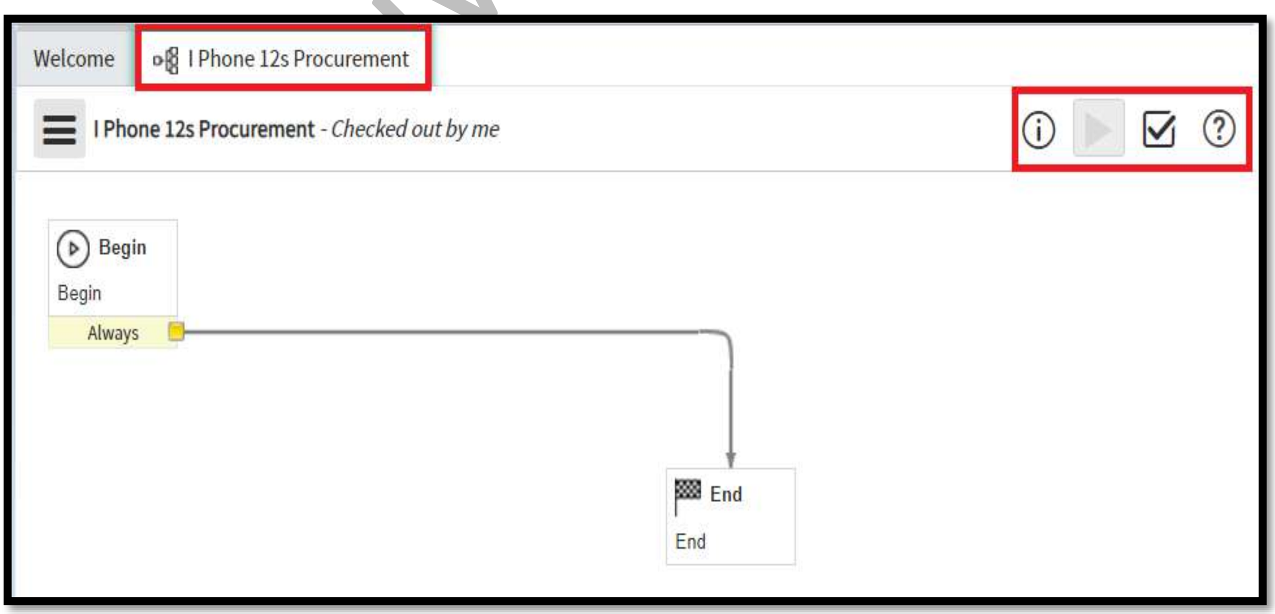


4. Fill new workflow details





5. **Name:** I Phone 12 Procurement
6. **Table:** Requested Item [sc_req_item]
7. **Description:** I Phone 12s Procurement
8. Open **Stages** Section
9. **Stage rendering:** Workflow-driven
10. **Stage Order:** Computed
11. Click on **Submit**



12. When we crated a new workflow by default, start and end activities will have generate like above

Exercise: 2 Add Activities to our workflow

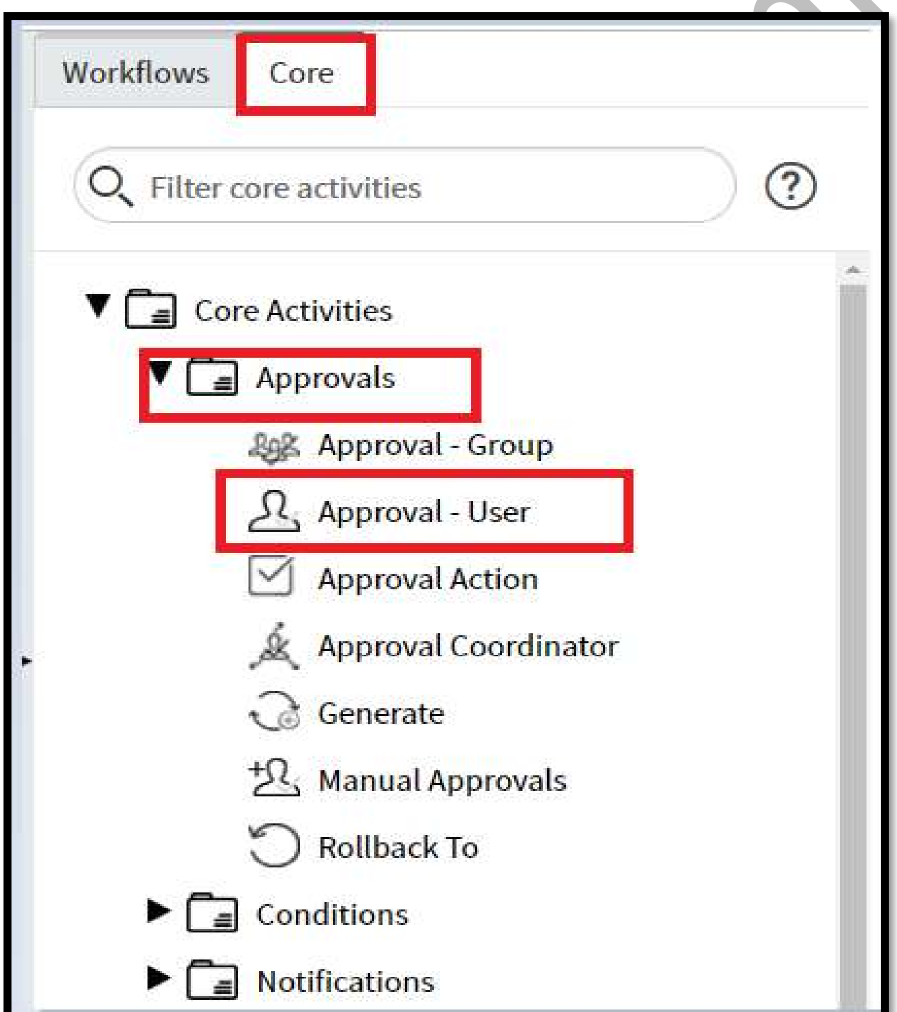
We can add required activities to our workflow to perform tasks such as **record insert, sending notifications,** or **requesting approvals.** Activities can succeed or fail, which can result in actions performed by other activities.

Add Approval Activity

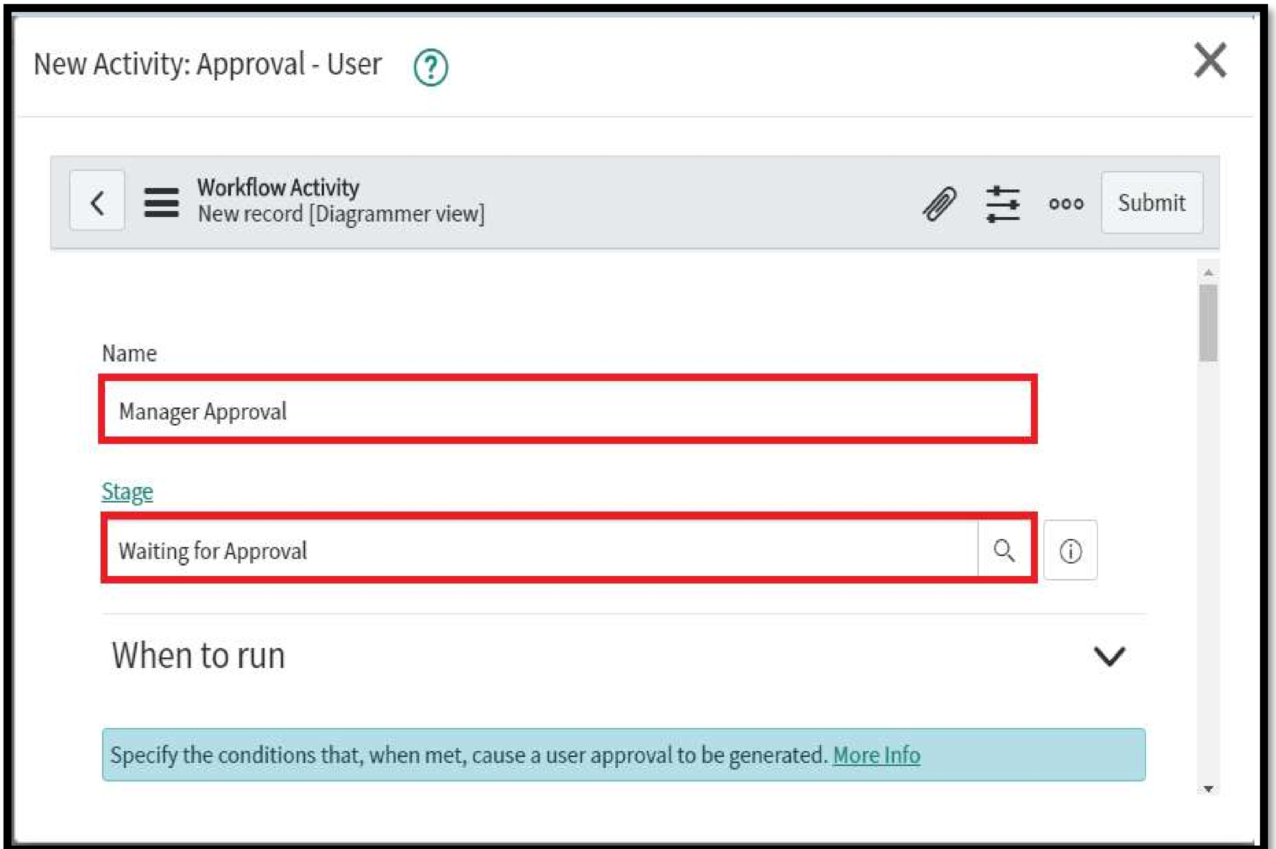
We have to fill the following steps to add an activity that requests **manager approval.**

Procedure

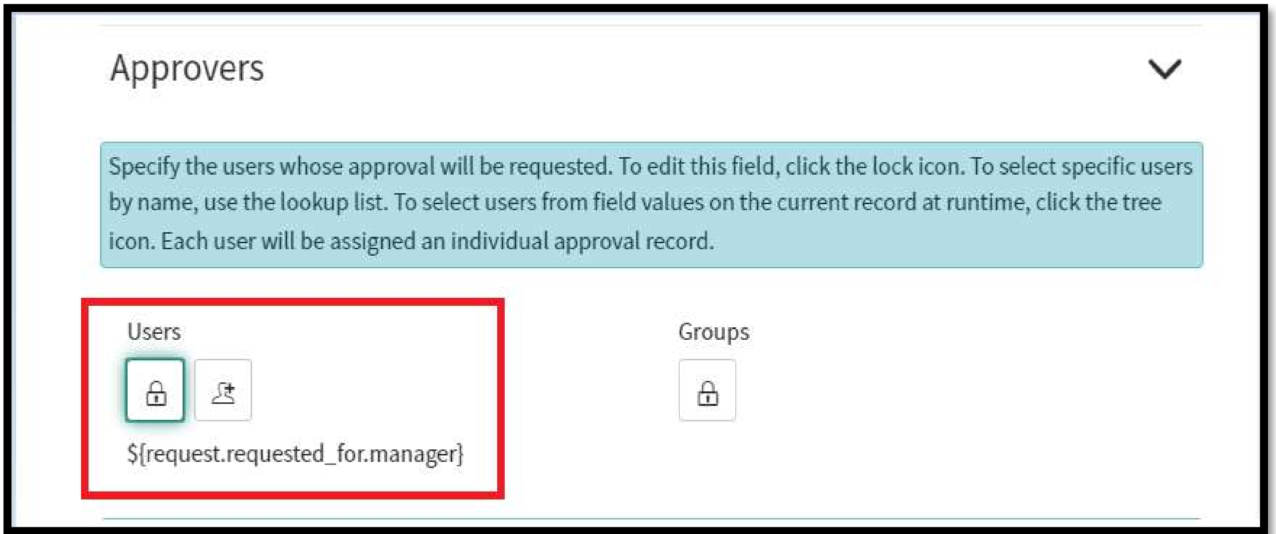
1. In the Core pallet, expand **Approvals** and drag the activity **Approval - User** into the arrow between **Begin** and **End activity.**



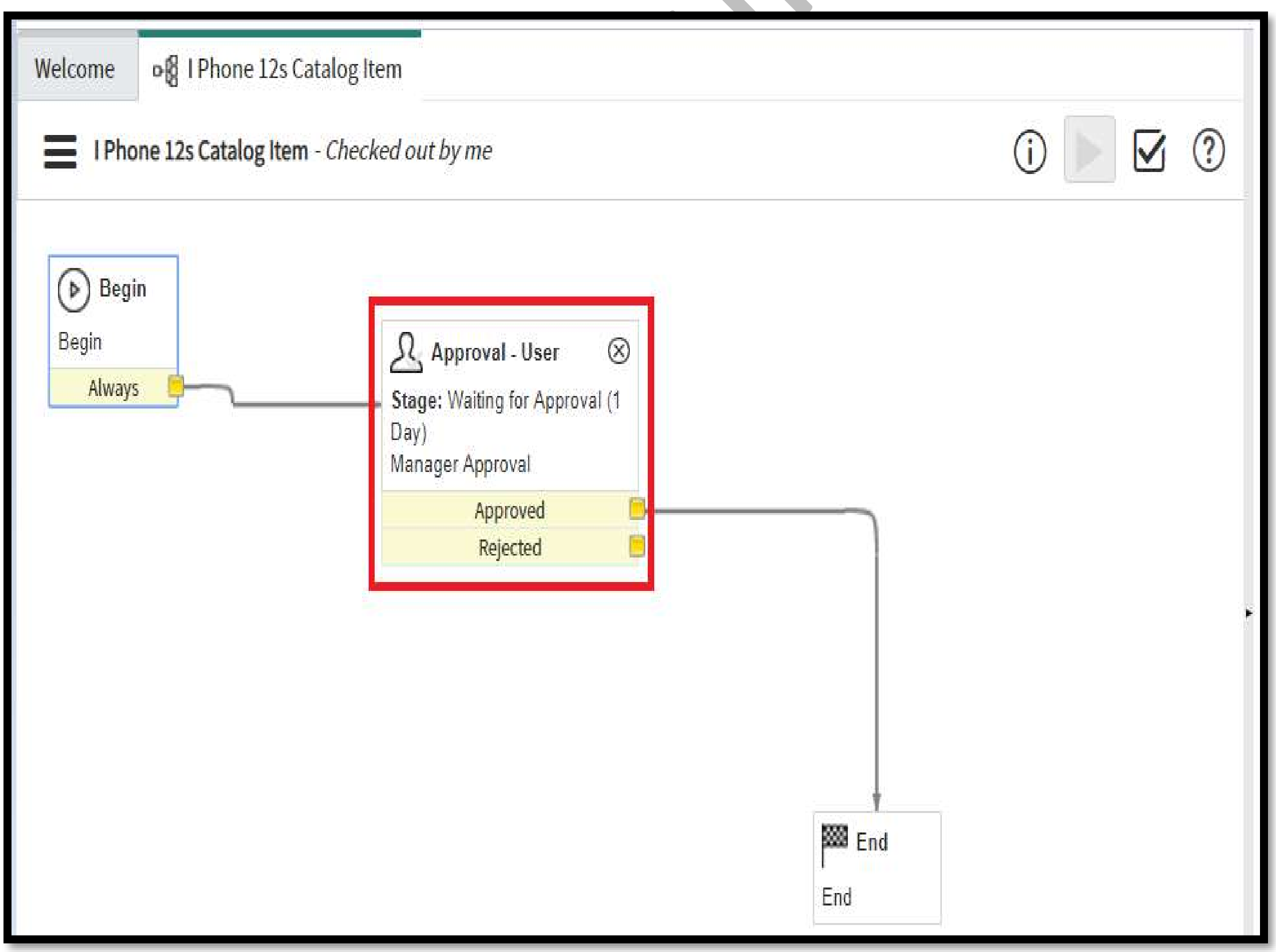
2. This activity generates an approval from the manager of the person requesting **I phone 12 mobile**
3. Fill form with the following information.
4. **Name:** Manager Approval
5. **Stage:** Waiting for Approval



6. **User:** Click the **lock** icon in the Approvers section, and then click the select fields icon. Expand **Request > Requested for > Manager**.



7. Click on **Submit**
8. Consider below screen shot about **Approval User** activity

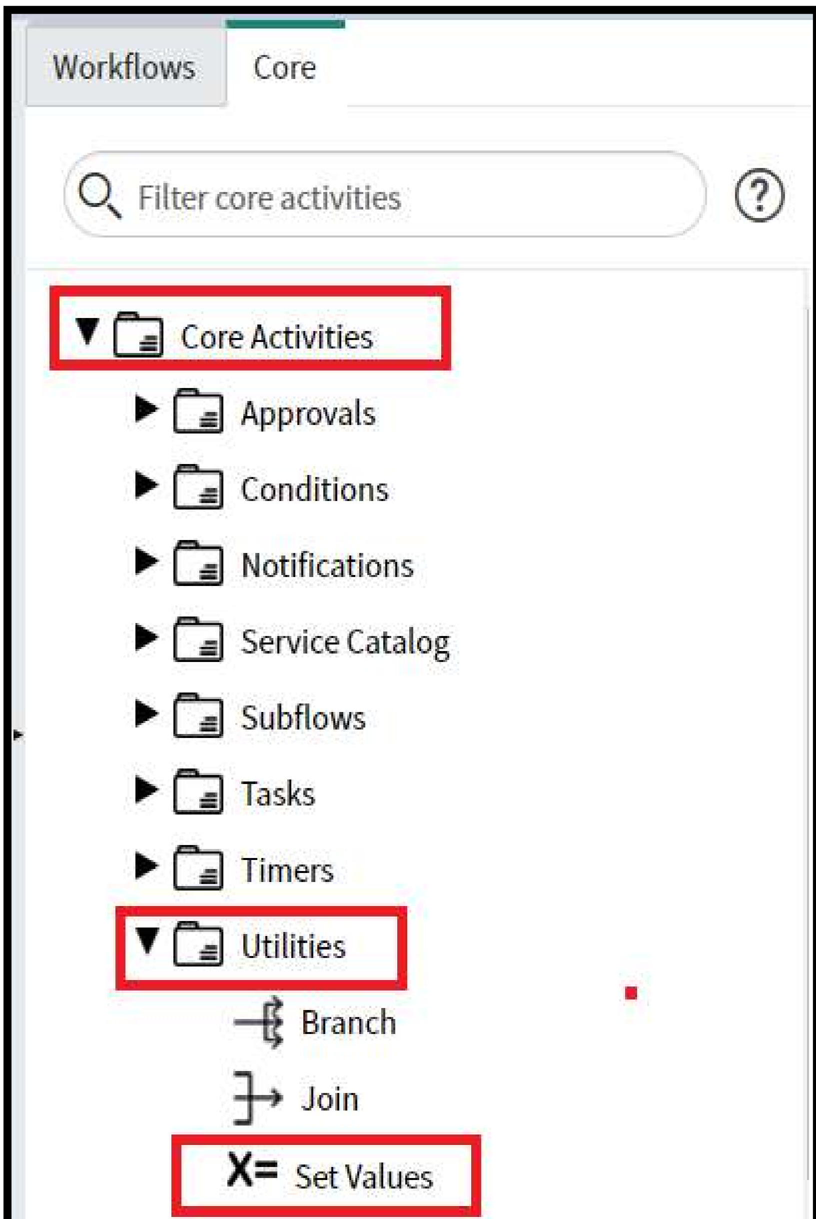


Add an action for when the approver rejects the request.

We are going to complete the following steps to add an action for when the approver **rejects** the request.

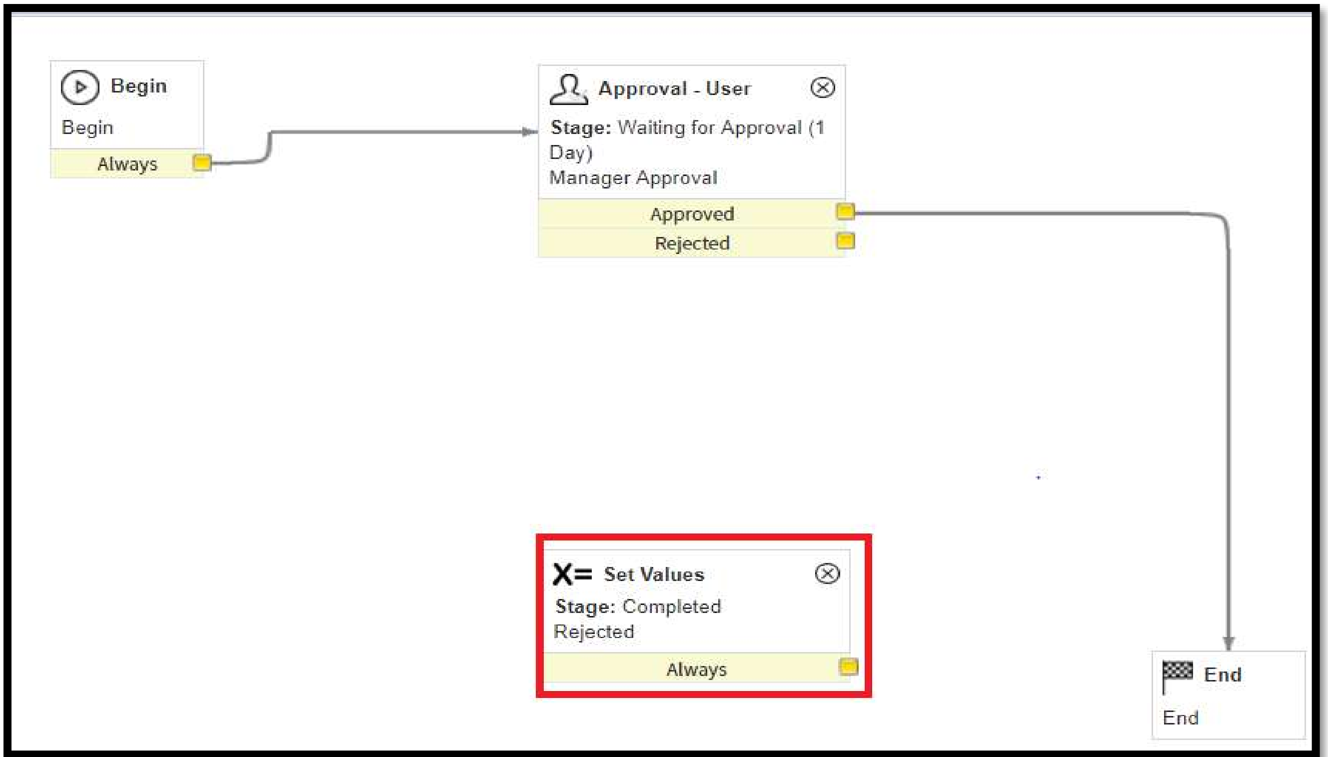
Procedure

1. Expand **Utilities** in the Core pane and drag the activity **Set Values** to the space below the **approval.**

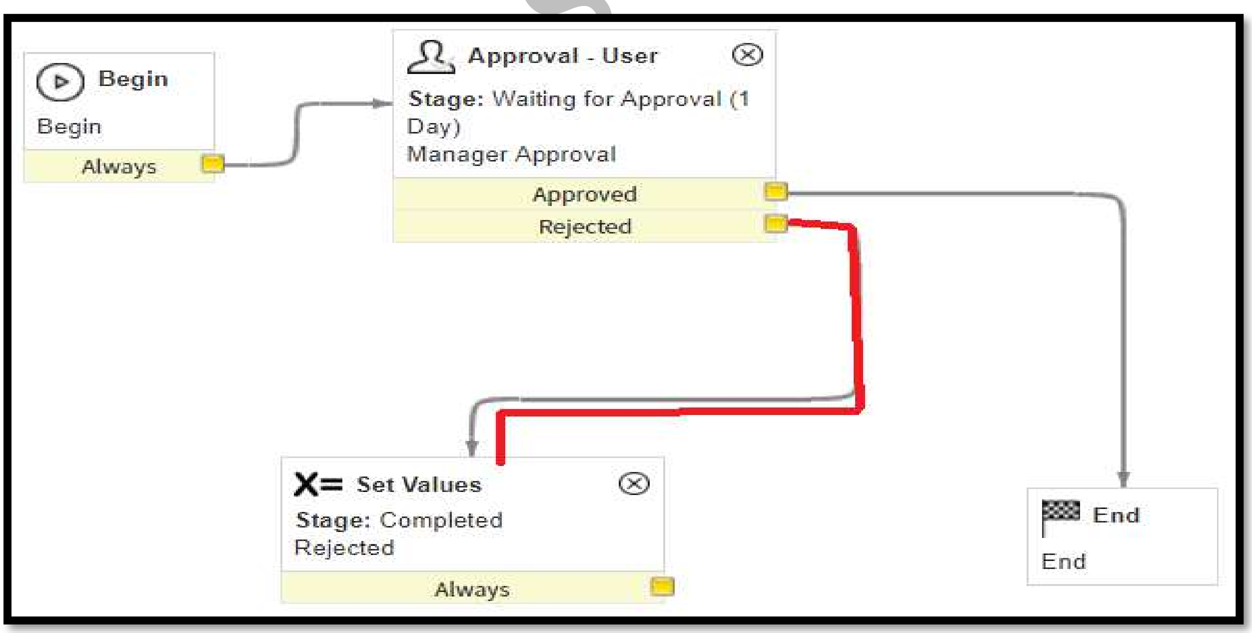


2. Complete the form with the following information.

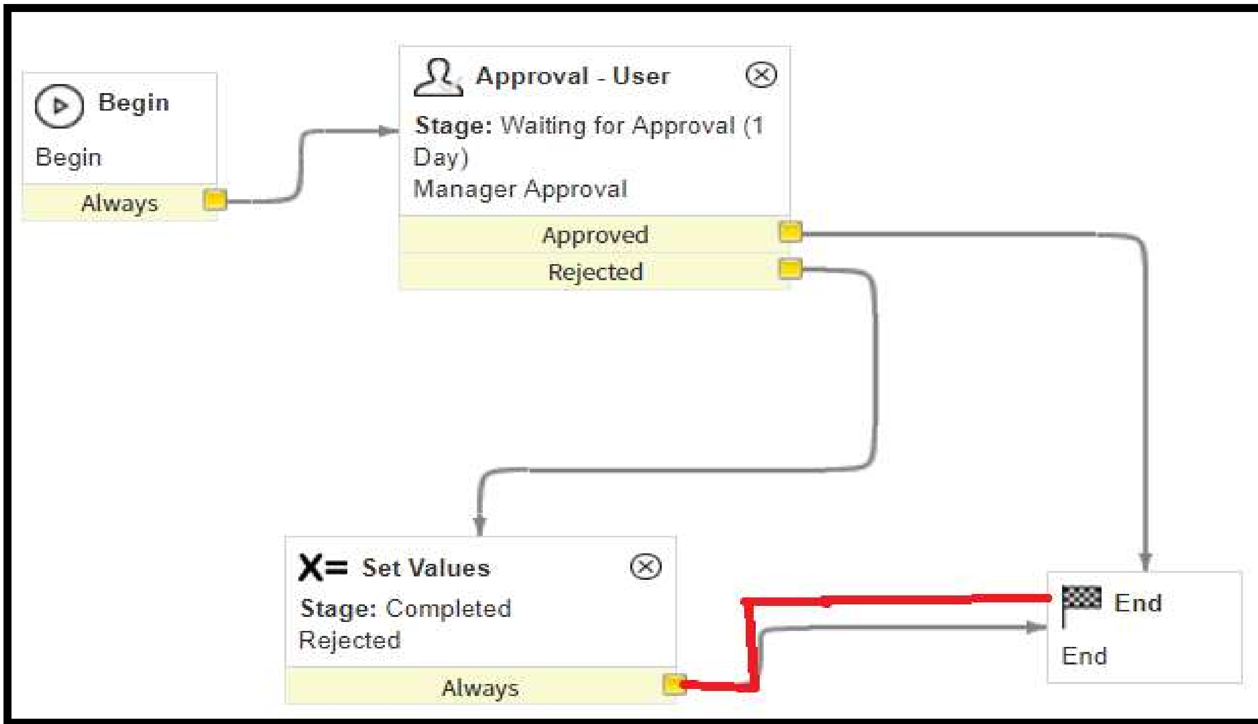
3. **Name:** Rejected
4. **Stage:** Completed
5. **Values Section**: [Approval] [Rejected]
6. Click on **Submit**



7. Drag from the yellow box beside **Rejected** on the **Approval - User** activity to the new **Set Values** activity.



8. Drag again from the yellow box beside **Always** on the **Set Values** activity to **End.**



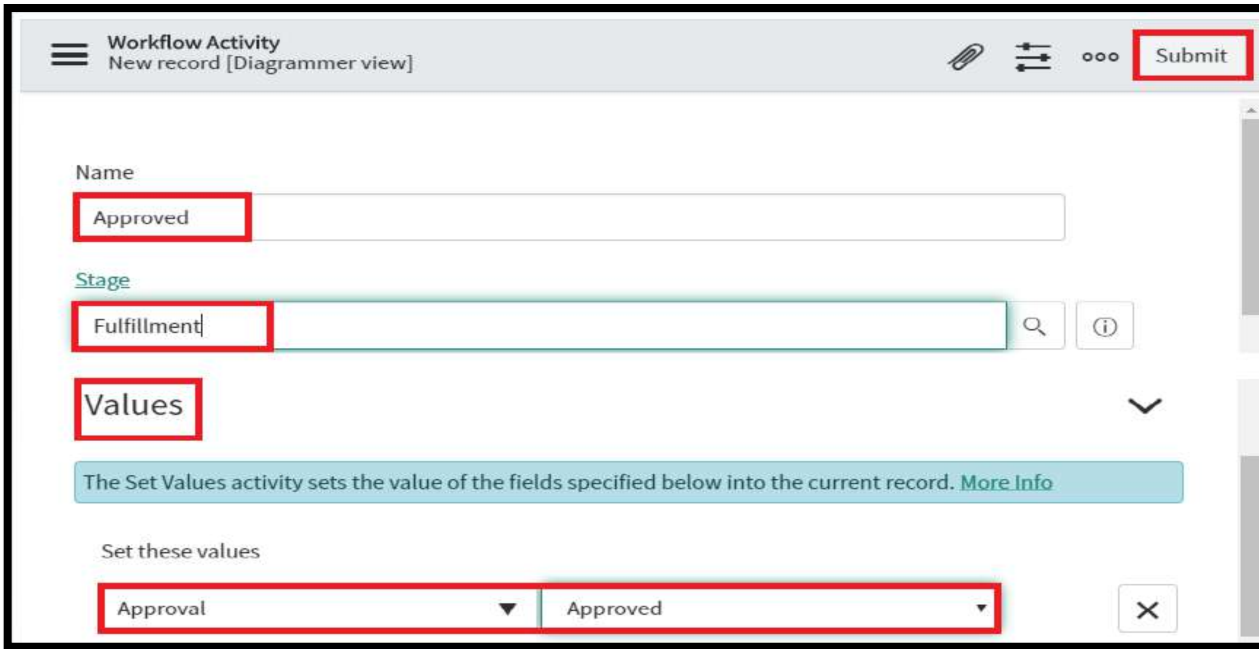
Note: This activity marks the request as **rejected** if the manager **rejects** the request and then ends the workflow

Add an action for when the approver Approved the request.

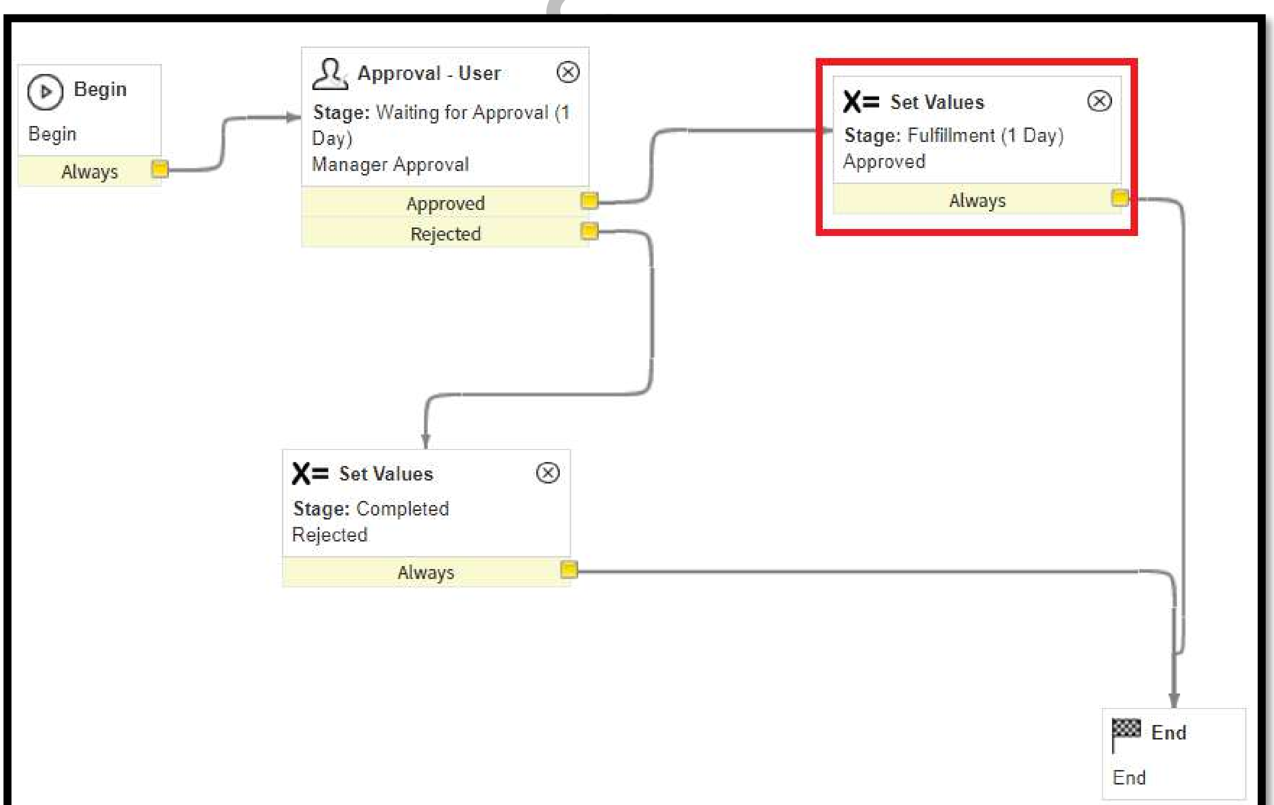
We are going to complete the following steps to add an action for when the approver **Approved** the request.

Procedure

1. Drag the activity **Set Values** onto the arrow between **Approval - User** and **End.**
2. Complete the form with the following information.



3. **Name:** Approved
4. **Stage:** Fulfillment
5. **Values:** [Approval] [Approved]
6. Click on **Submit**



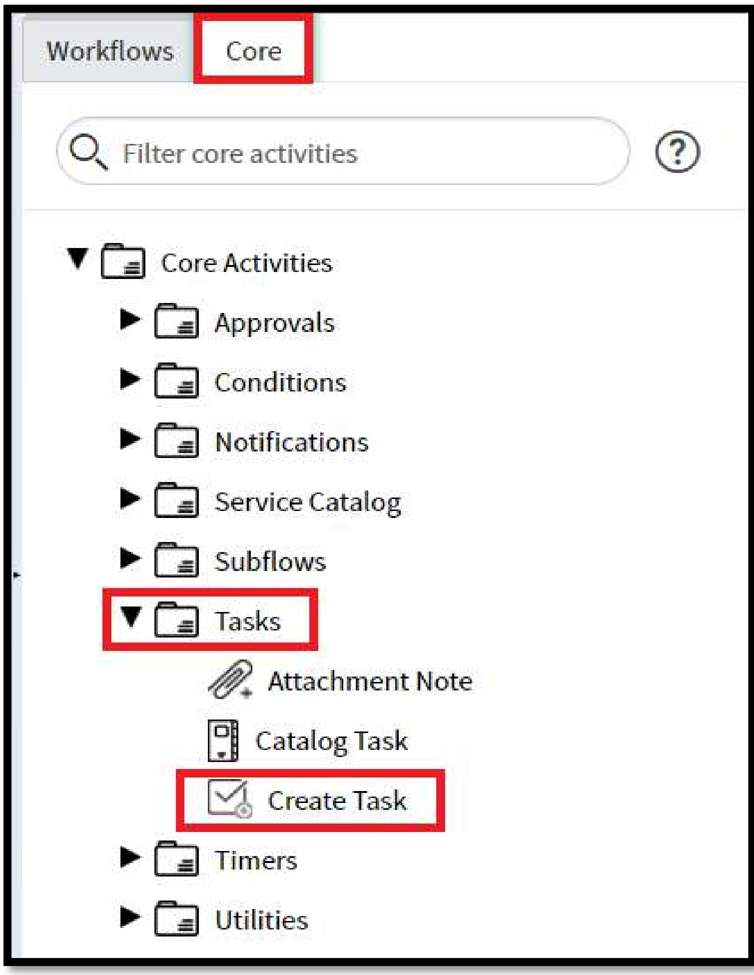
Note: This activity marks the request as approved once a manager approves it.

Add Task Activity

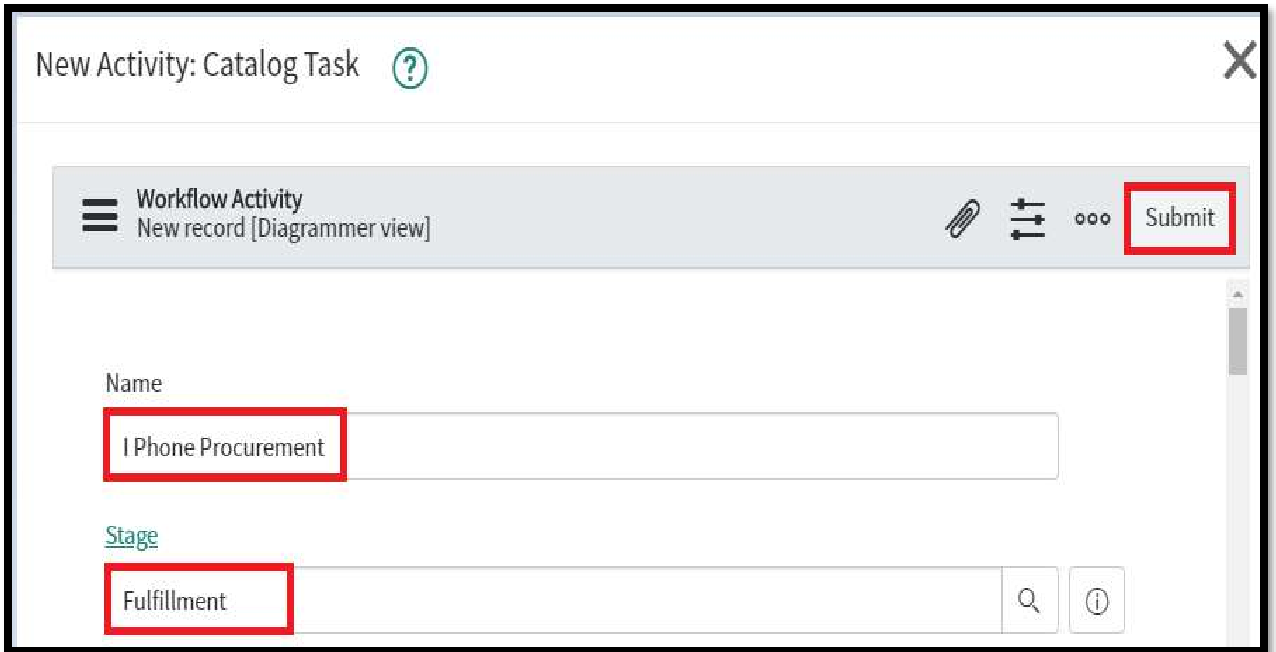
This task activity will be defining Procurement to procure a I phone mobile for the user.

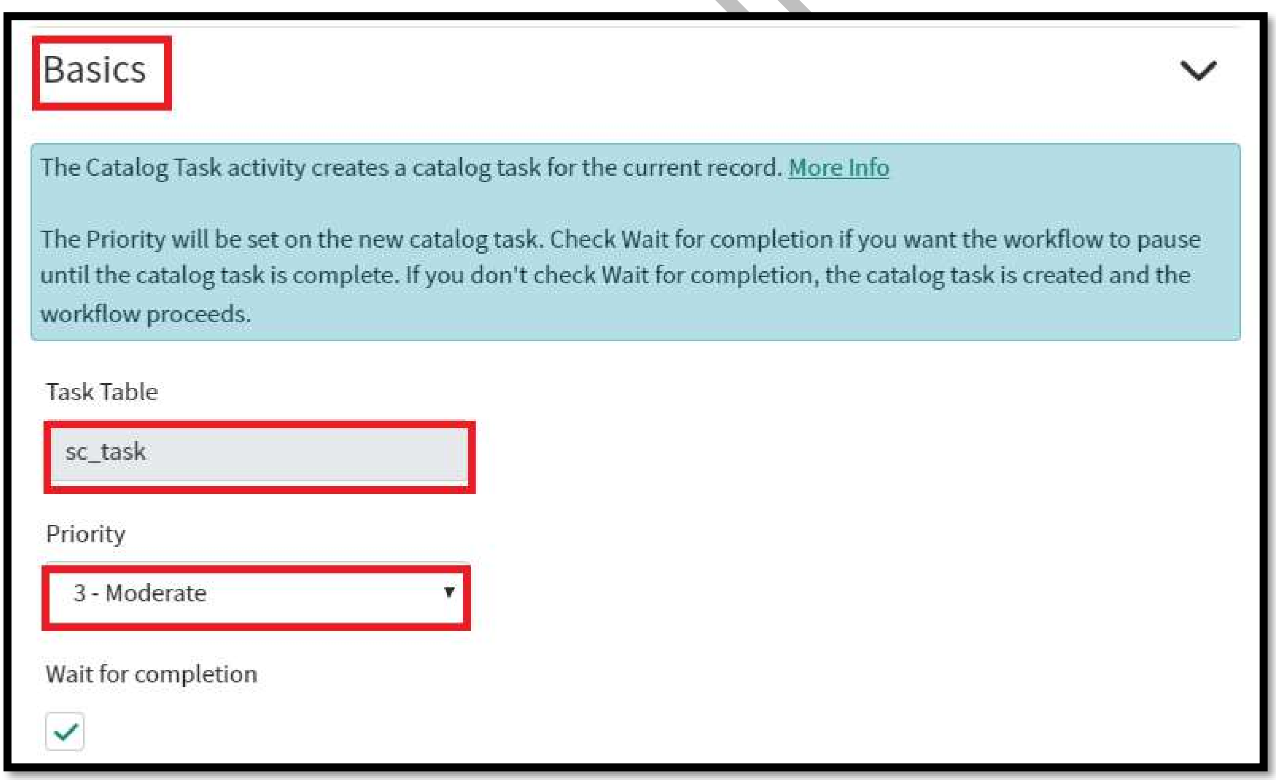
Procedure

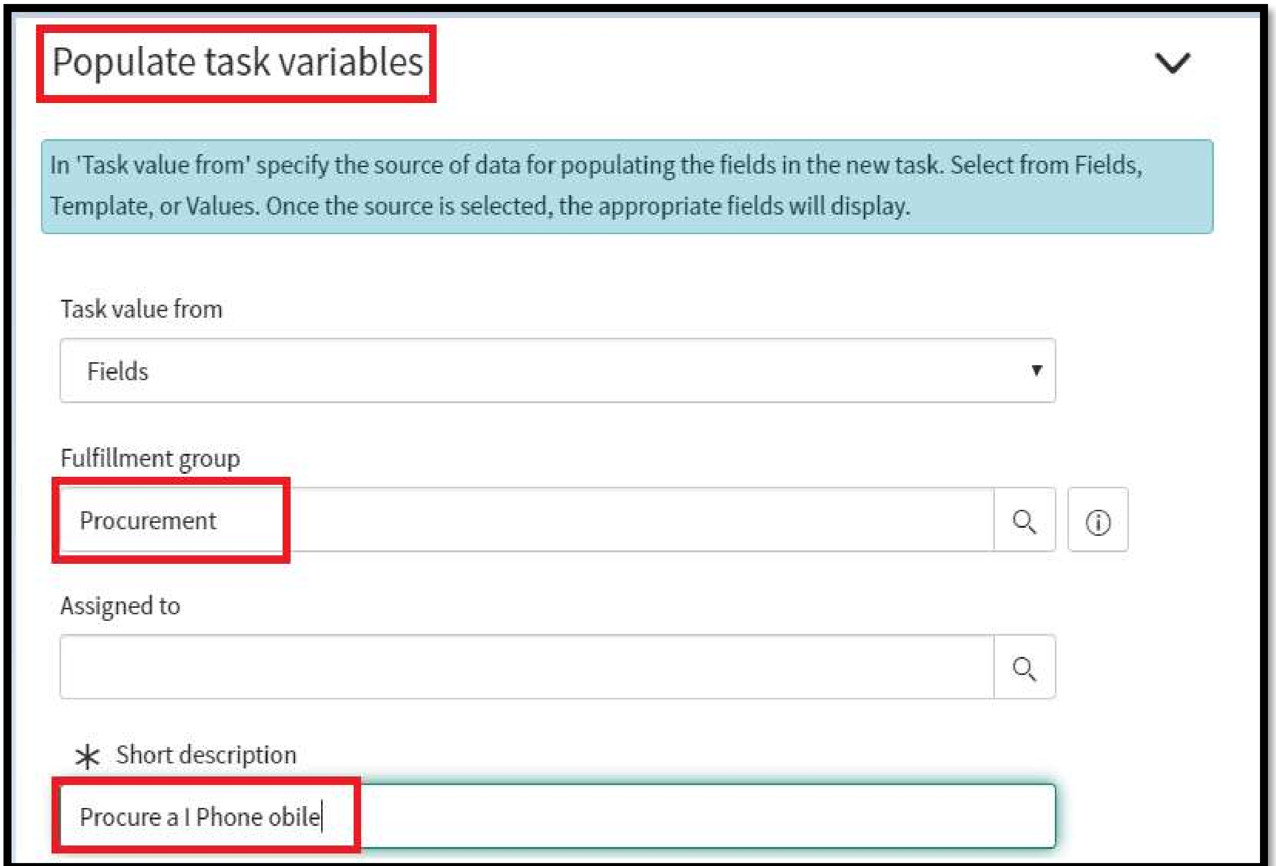
1. Expand Tasks and drag the activity Create Task onto the arrow between Approval - User and End,



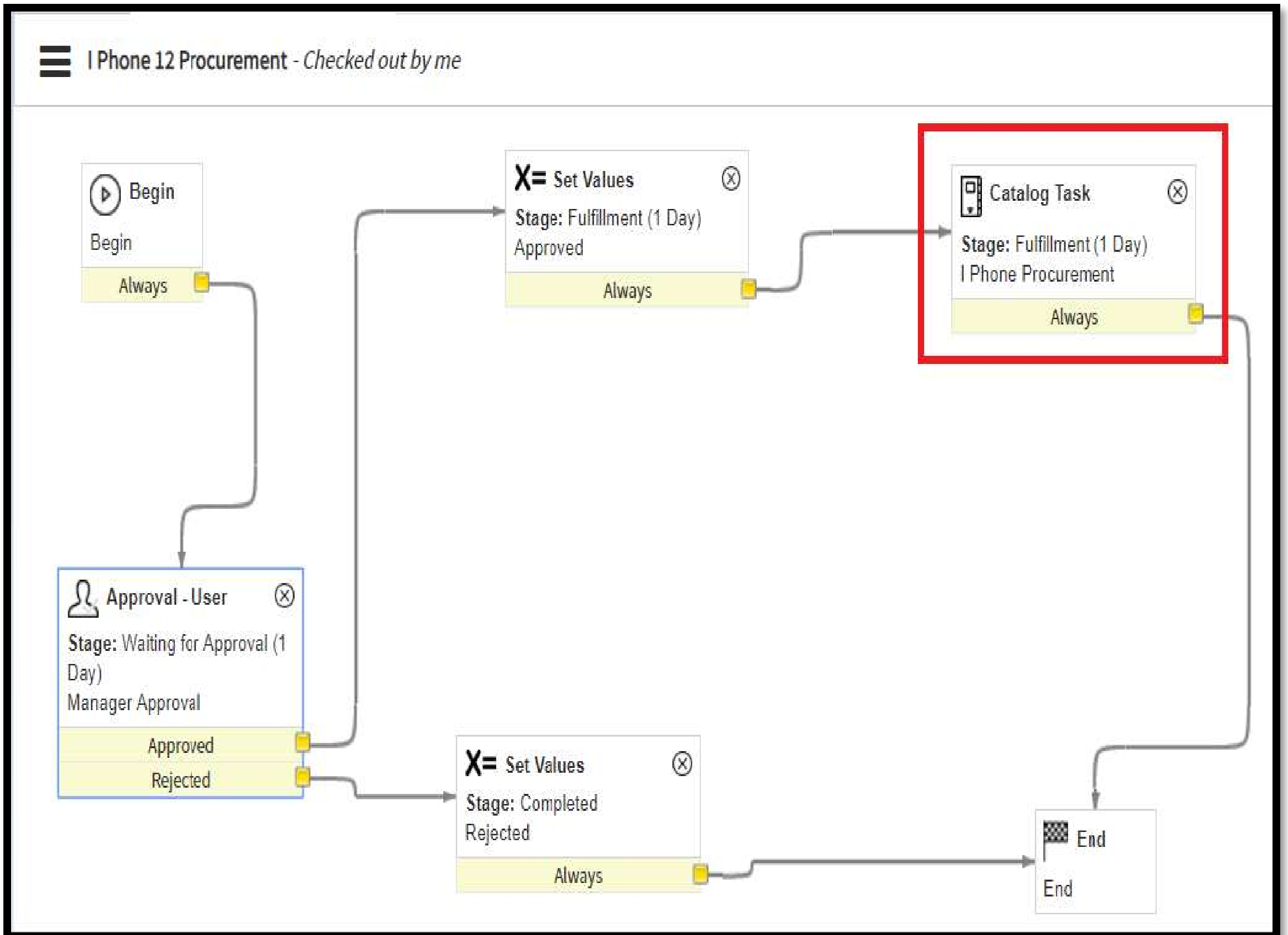
2. Fill the form with following information


New Activity: Catalog Task
Workflow Activity
New record [Diagrammer view]
Submit
Name
I Phone Procurement
Stage
Fulfillment


Basics
The Catalog Task activity creates a catalog task for the current record. More Info
The Priority will be set on the new catalog task. Check Wait for completion if you want the workflow to pause until the catalog task is complete. If you don't check Wait for completion, the catalog task is created and the workflow proceeds.
Task Table
sc_task
Priority
3 - Moderate
Wait for completion



3. **Name:** I Phone Procurement
4. **Stage:** Fulfillment
5. **Task Type:** Catalog Task [sc_task]
6. **Priority**: **3 – Moderate**
7. **Fulfillment group**: Procurement
8. **Short description**: Procure a I phone 12s Mobile
9. Click on **Submit**

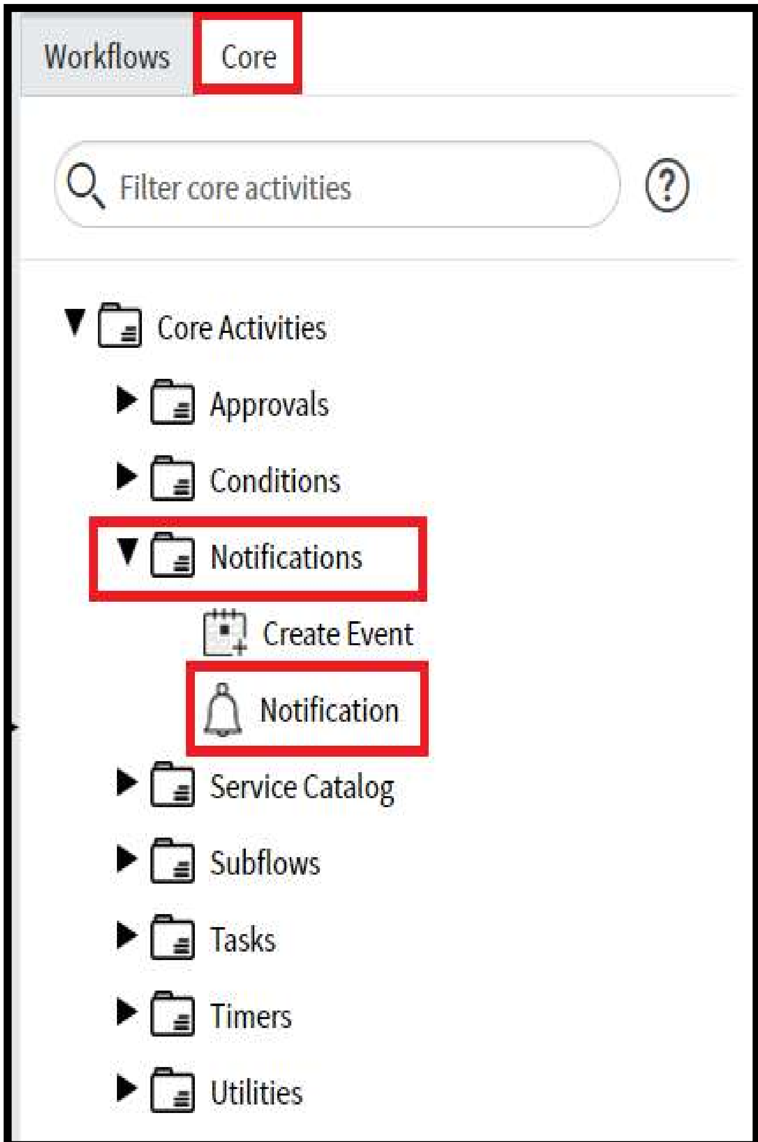


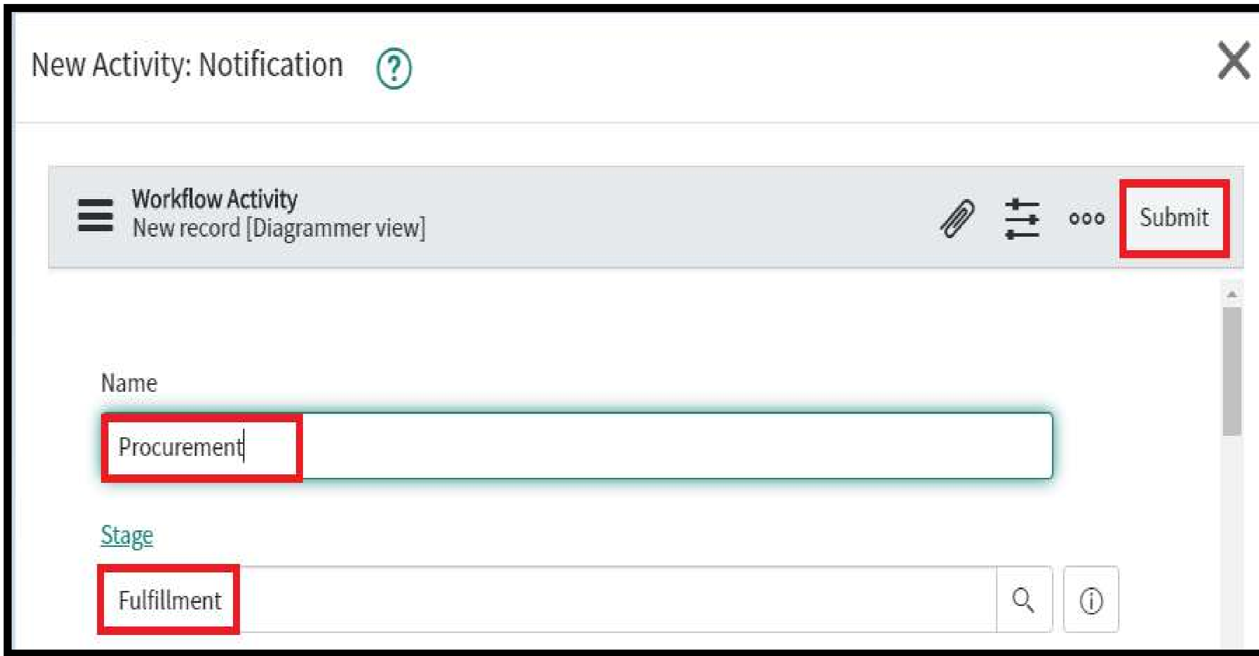
Add Notifications Activity

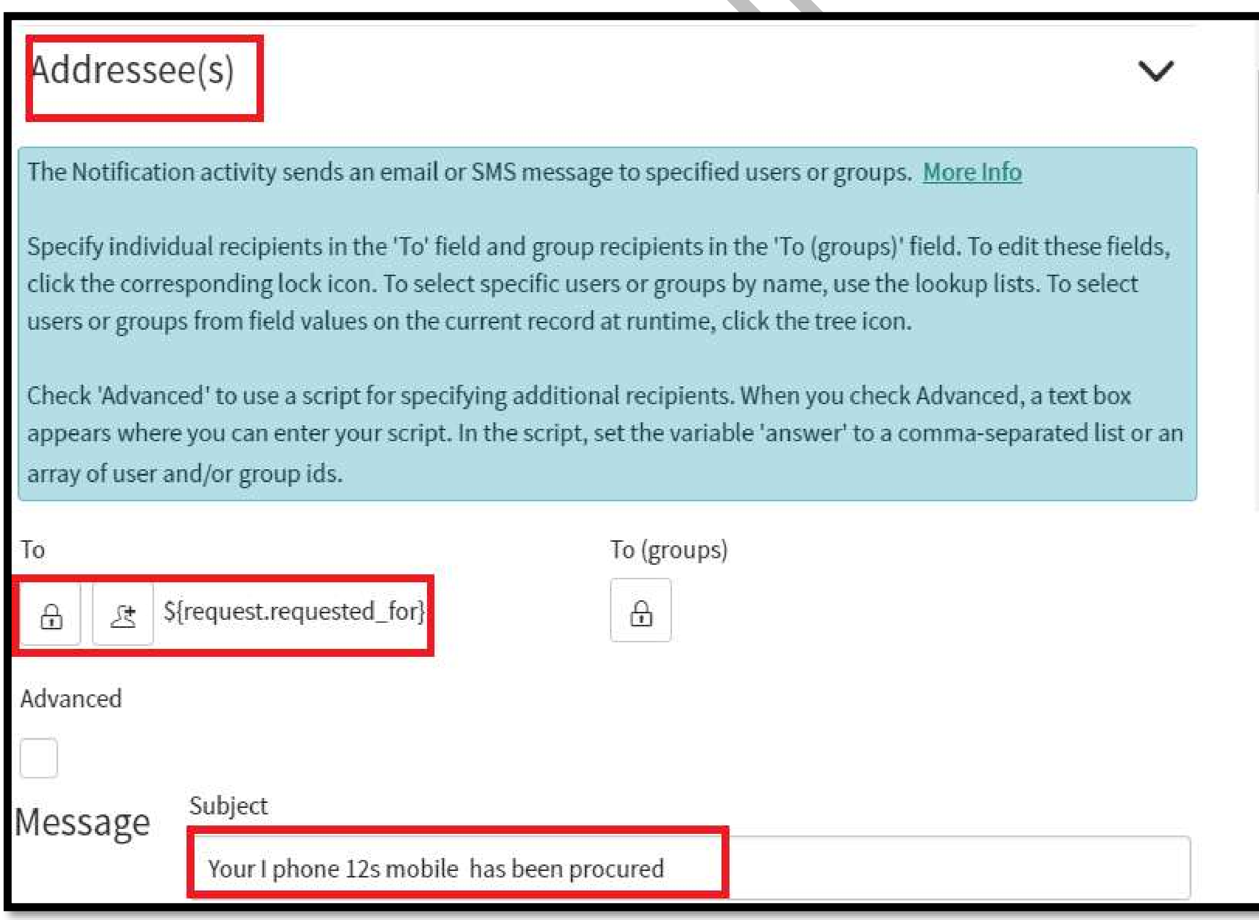
This activity used to mark the request as approved if a manager approves it then send the notification to customer.

Procedure

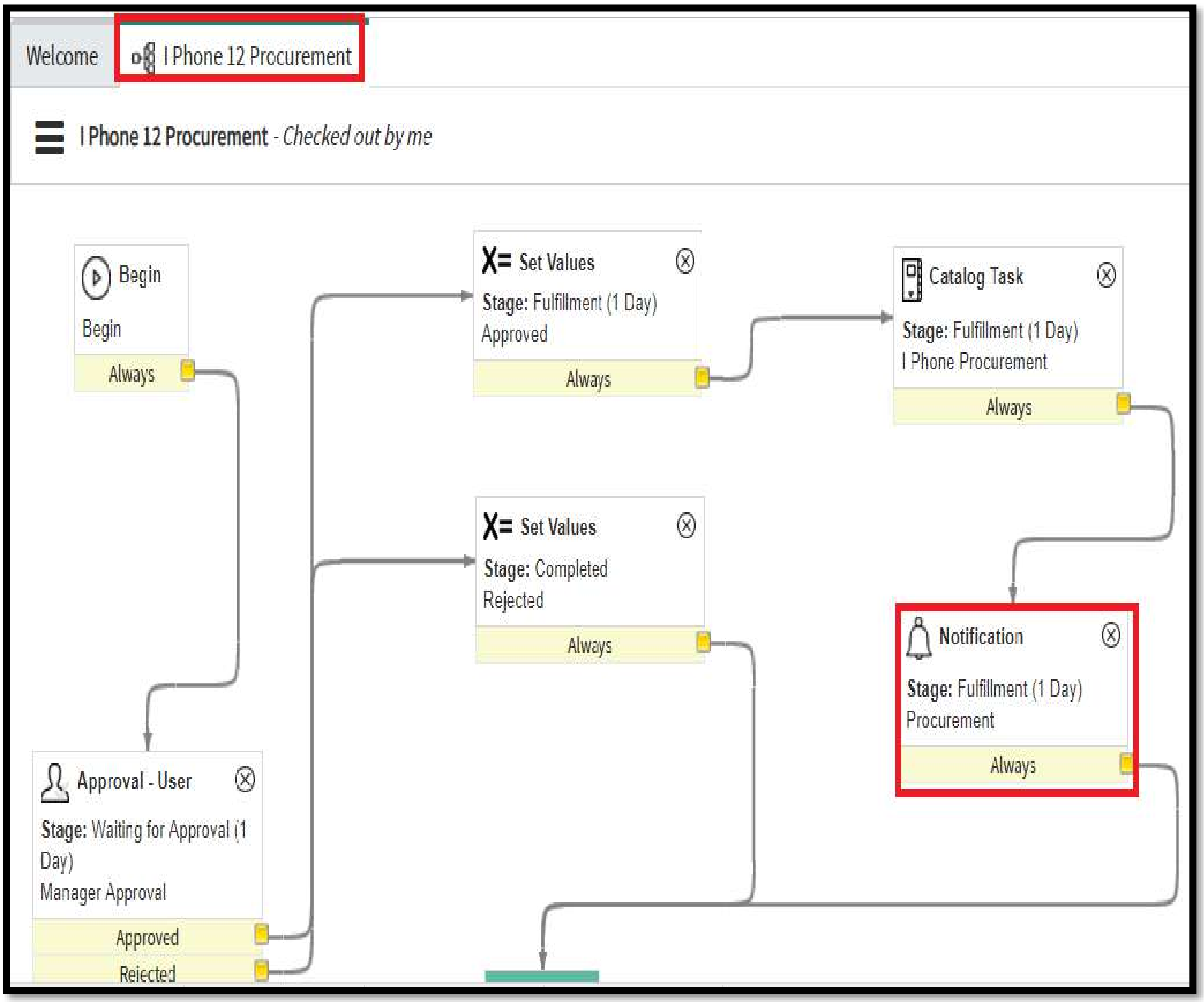
1. Expand Notifications and drag the activity Notification onto the arrow between I Phone 12 s Procurement and End,
2. Fill the following information


Workflows
Core
Filter core activities
Core Activities
Approvals
Conditions
Notifications
Create Event
Notification
Service Catalog
Subflows
Tasks
Timers
Utilities


New Activity: Notification
Workflow Activity
New record [Diagrammer view]
Submit
Name
Procurement
Stage
Fulfillment


Addressee(s)
The Notification activity sends an email or SMS message to specified users or groups. More Info
Specify individual recipients in the 'To' field and group recipients in the 'To (groups)' field. To edit these fields, click the corresponding lock icon. To select specific users or groups by name, use the lookup lists. To select users or groups from field values on the current record at runtime, click the tree icon.
Check 'Advanced' to use a script for specifying additional recipients. When you check Advanced, a text box appears where you can enter your script. In the script, set the variable 'answer' to a comma-separated list or an array of user and/or group ids.
To
${request.requested_for}
To (groups)
Advanced
Message
Subject
Your I phone 12s mobile has been procured

3. **Name:** Procurement
4. **Stage:** Fulfillment
5. **Address to:** Click the **lock** icon in the Addressee(s) section, and then click the select fields icon. Expand **Requested** and **select Requested for**.
6. **Subject: Your I phone 12s mobile** has been procured.
7. Click on **Submit**

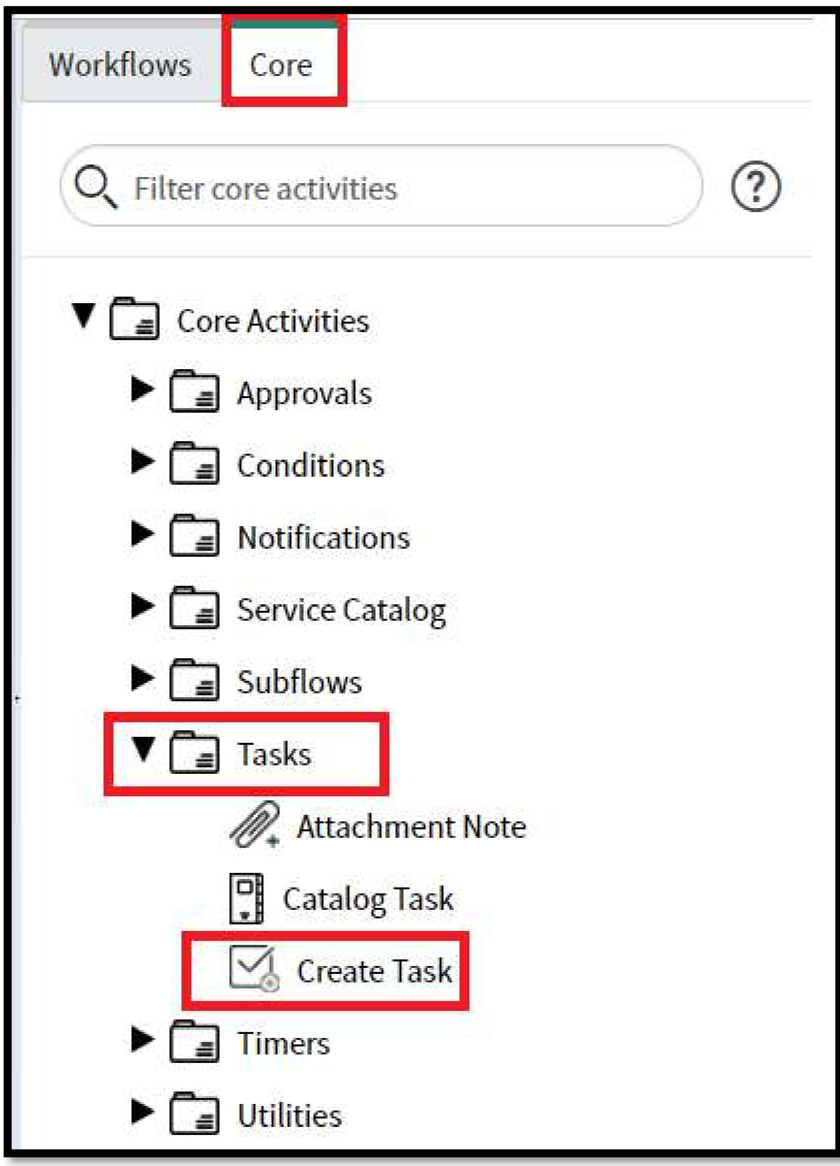


Add Task Activity

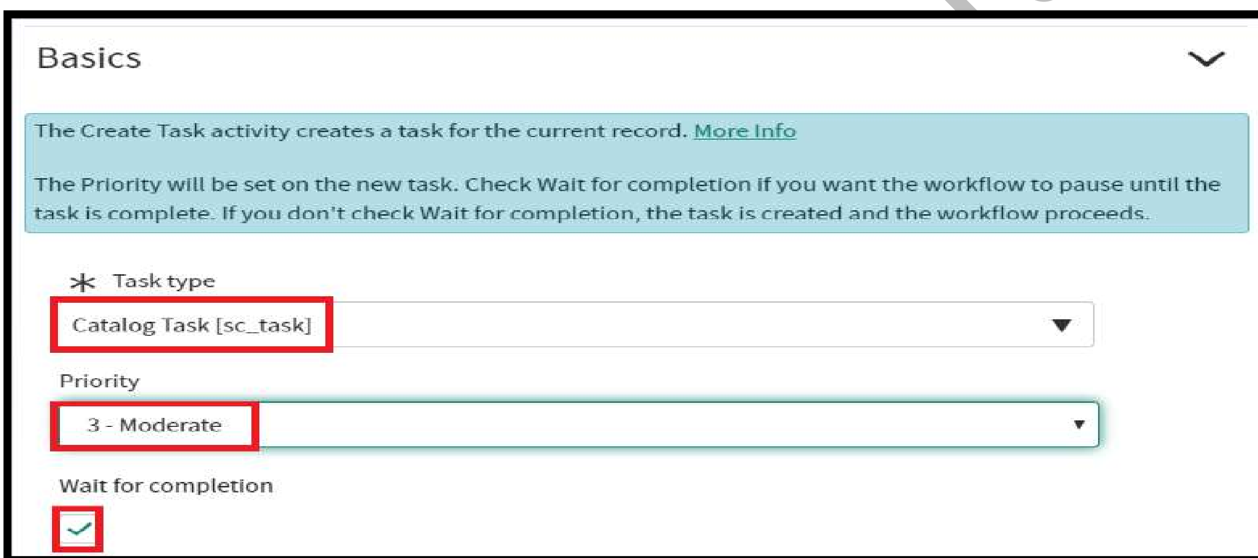
This task activity will be defining Hardware to configure the I phone 12s mobile and prepare it for use.

Procedure

1. Drag the activity **Create Task** onto the arrow between **I phone 12 procurement** and **End**,
2. Fill the following information below

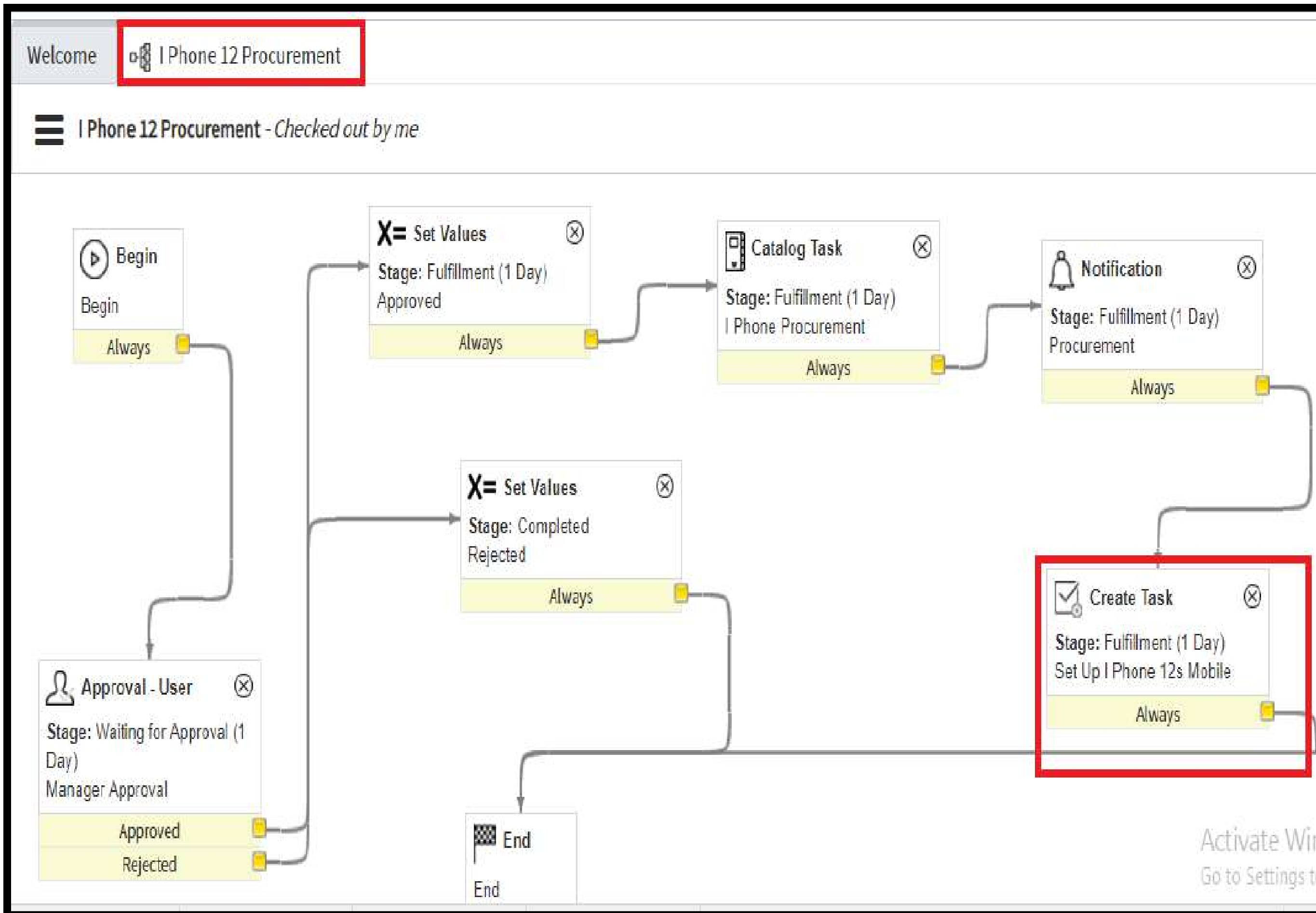



New Activity: Create Task
Workflow Activity
New record [Diagrammer view]
Submit
Name
Set Up I Phone 12s Mobile
Stage
Fulfillment


Basics
The Create Task activity creates a task for the current record. More Info
The Priority will be set on the new task. Check Wait for completion if you want the workflow to pause until the task is complete. If you don't check Wait for completion, the task is created and the workflow proceeds.
Task type
Catalog Task [sc_task]
Priority
3 - Moderate
Wait for completion


Populate task variables
Fields
Fulfillment group
Hardware
Assigned to
Short description
Set up I Phone 12 mobile

3. **Name:** Set up I Phone 12 Mobile
4. **Stage:** Fulfillment
5. **Task Type:** Catalog Task
6. **Priority:** Moderate
7. **Fulfillment Group:** Hardware
8. **Short Description:** Set up I Phone 12s Mobile

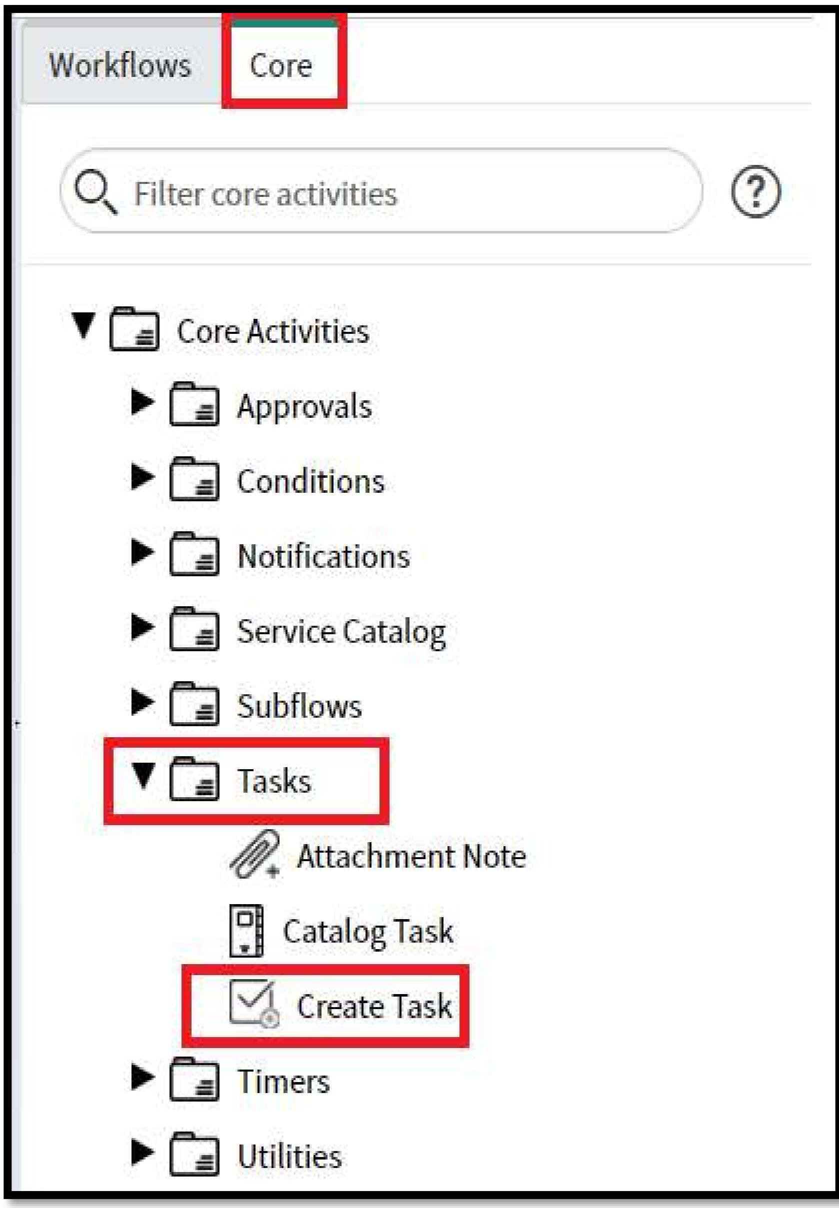


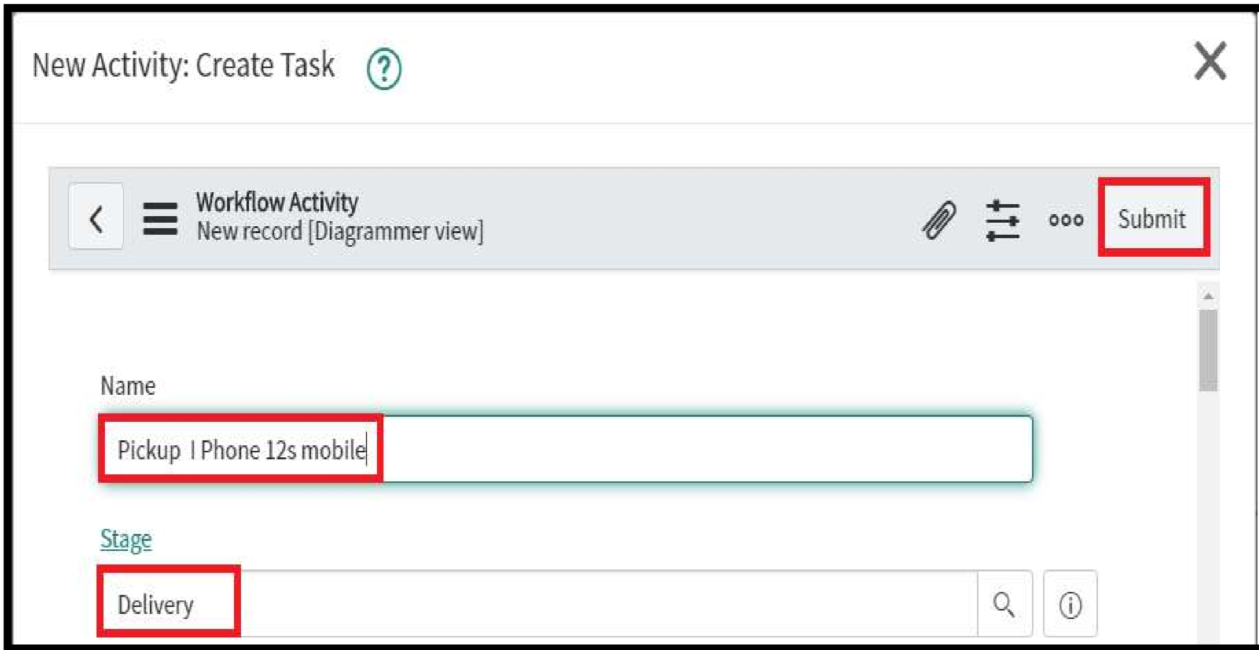
Add Task Activity

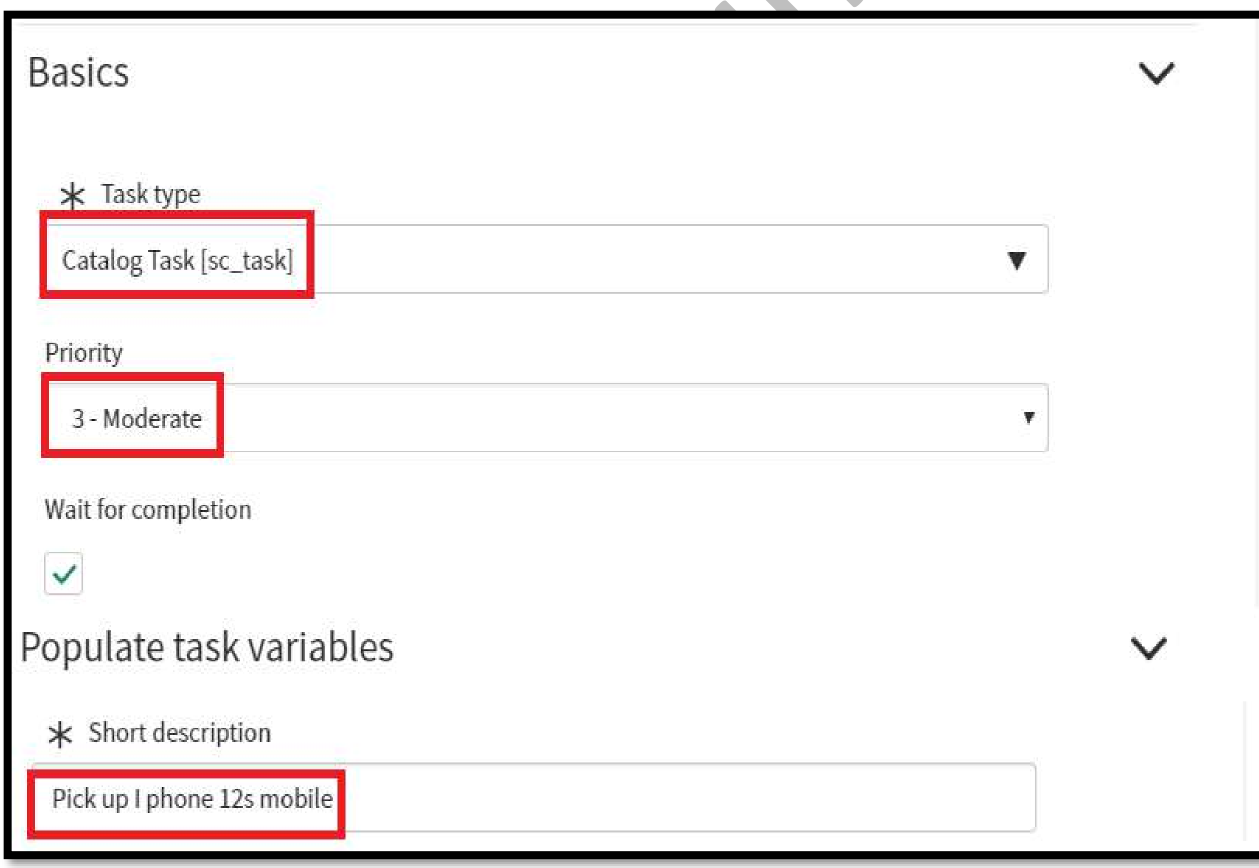
This task will be informing the requester to come pick up the **I phone 12s mobile.**

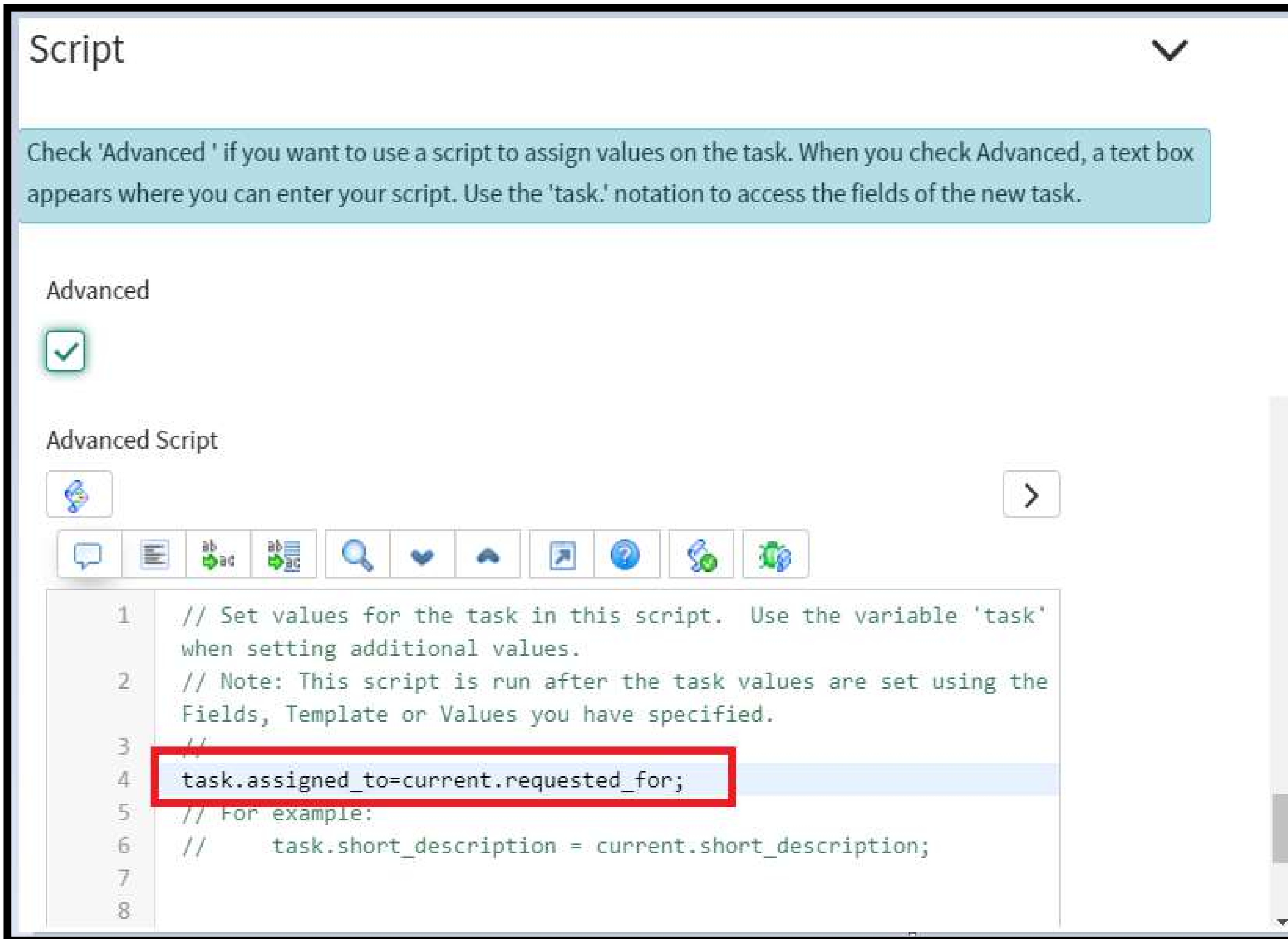
Procedure

1. Drag the activity Create Task onto the arrow between Set Up I Phone 12s mobile and End,
2. Fill the following information



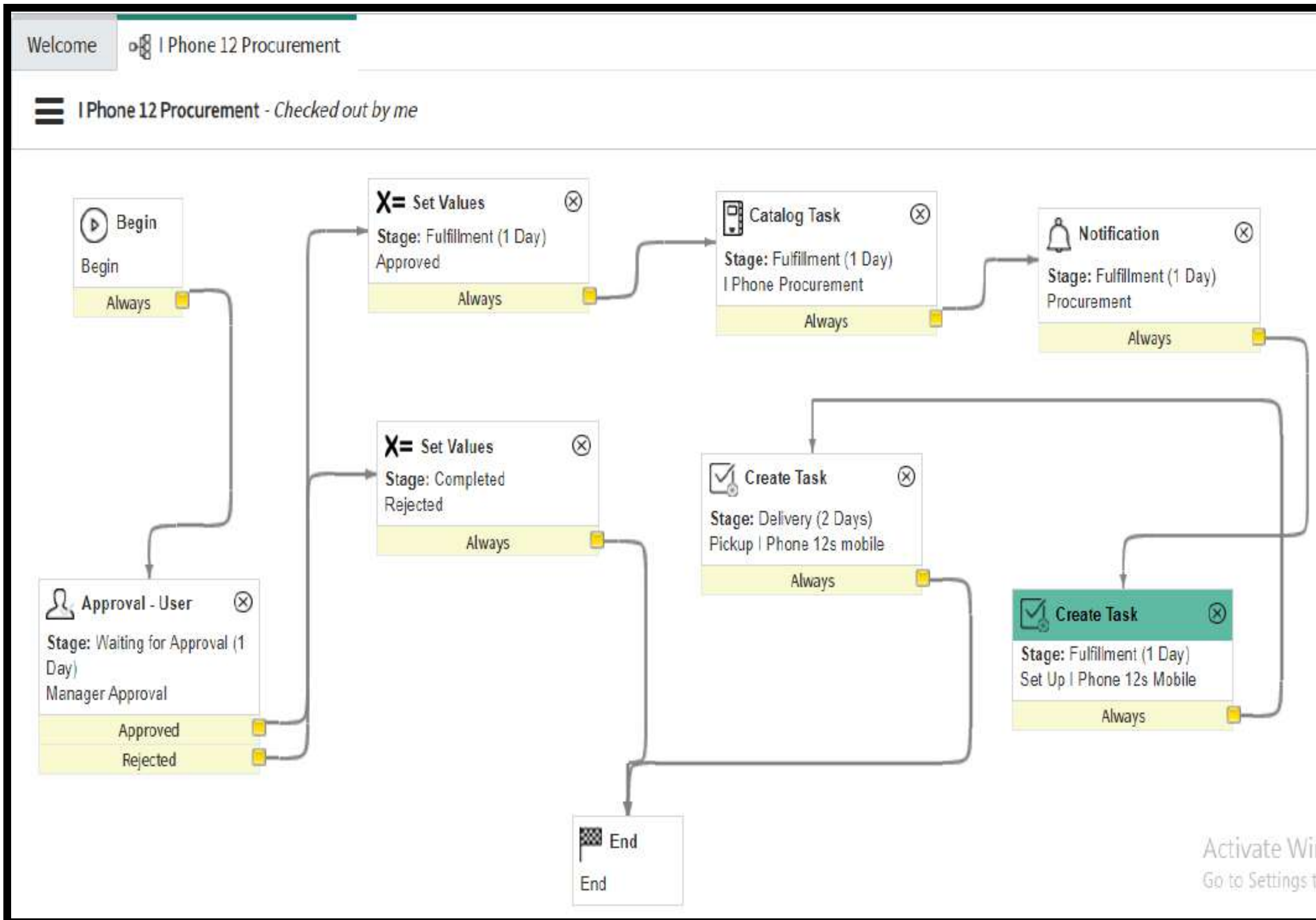

New Activity: Create Task
Workflow Activity
New record [Diagrammer view]
Submit
Name
Pickup I Phone 12s mobile
Stage
Delivery


Basics
Task type
Catalog Task [sc_task]
Priority
3 - Moderate
Wait for completion
Populate task variables
Short description
Pick up I phone 12s mobile



3. **Name:** I Phone 12s mobile
4. **Stage:** Delivery
5. **Task Type:** Catalog Task [sc_task]
6. **Priority:** 3 - Moderate
7. Advanced field assigns this task to the requester.
8. **Short description:** Pick up I Phone 12s mobile.
9. **Advanced:** Select the check box and set the Advanced Script value to

   task.assigned_to=current.requested_for;

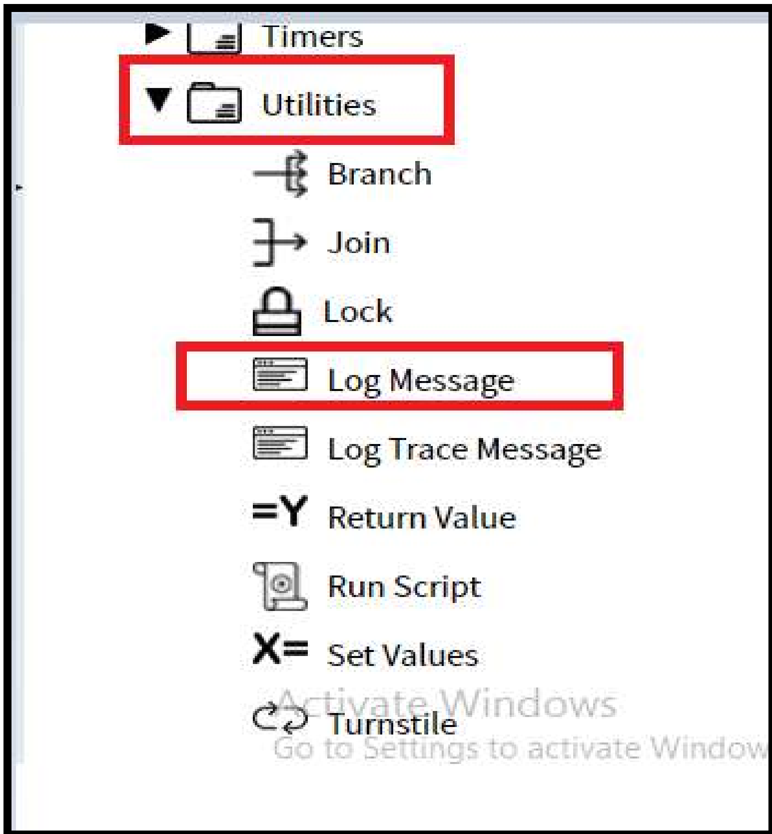


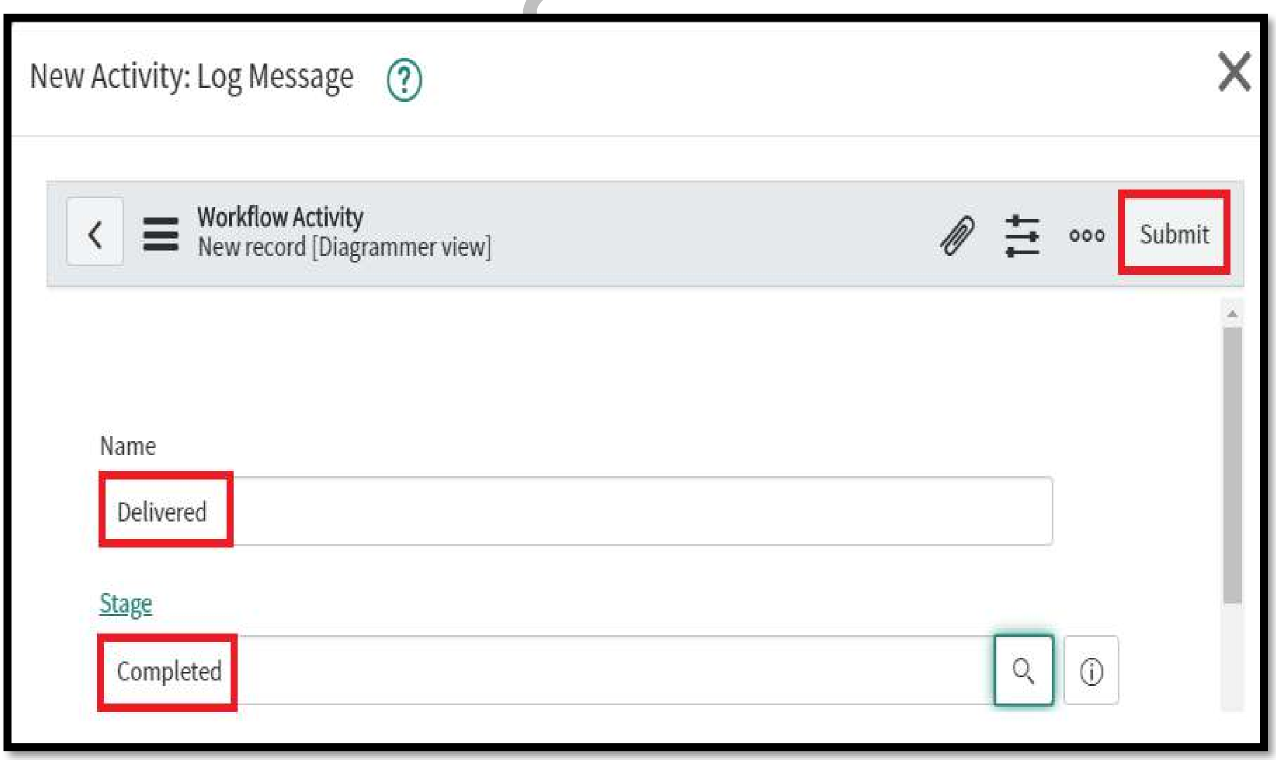
Add Utilities Activity

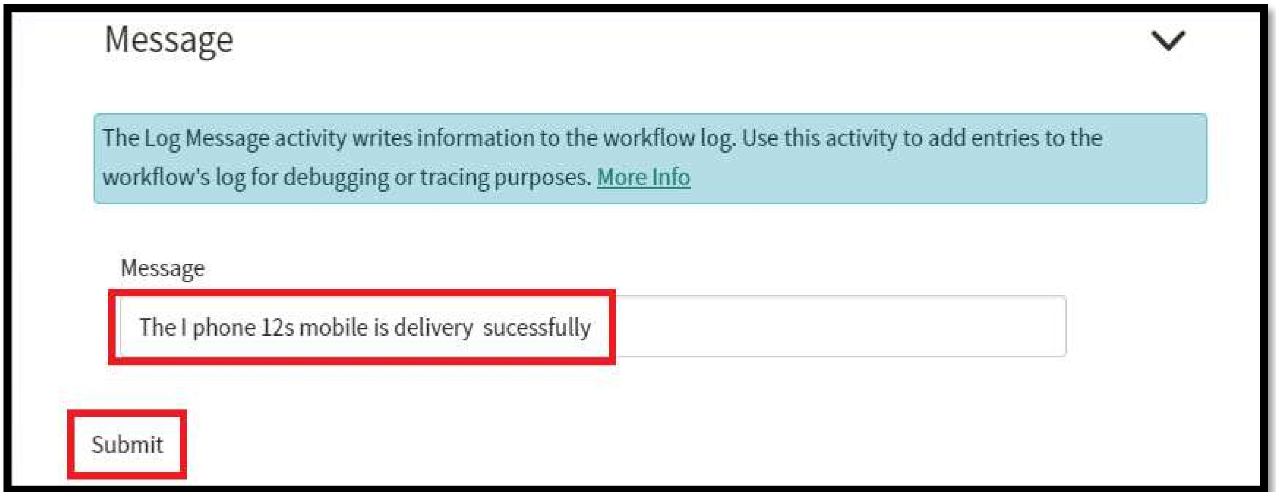
This activity uses to leaves a message in the log to record the successful completion of the workflow.

Procedure

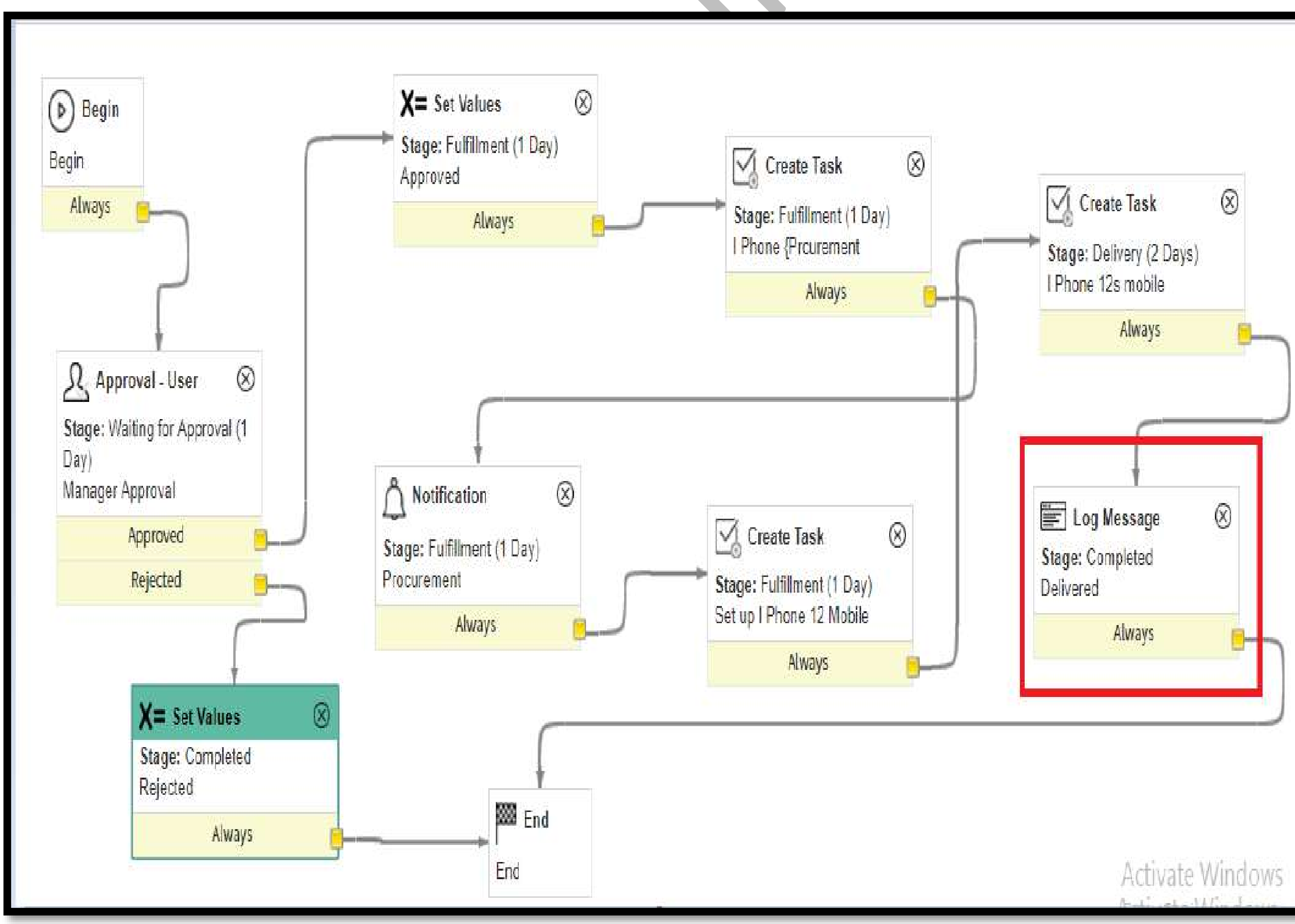
1. Drag the activity **Log Message** onto the arrow between **Pick Up I Phone 12s mobile and End**,
2. Fill the following information


Timers
Utilities
Branch
Join
Lock
Log Message
Log Trace Message
Return Value
Run Script
Set Values
Turnstile
Activate Windows
Go to Settings to activate Windows


New Activity: Log Message
Workflow Activity
New record [Diagrammer view]
Submit
Name
Delivered
Stage
Completed



3. **Name**: Delivered
4. **Stage:** Completed
5. **Message:** The 12s mobile has been delivered successfully
6. Click on **Submit**

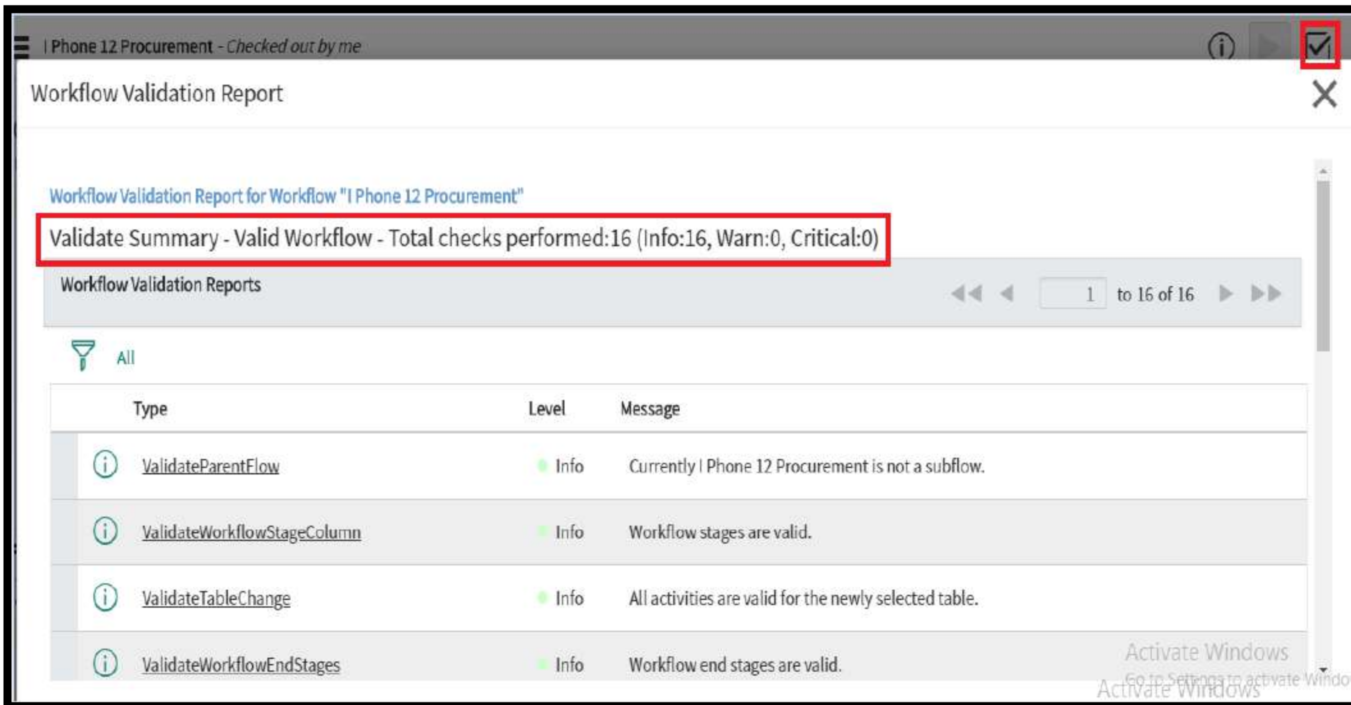


Validating and Publishing workflow

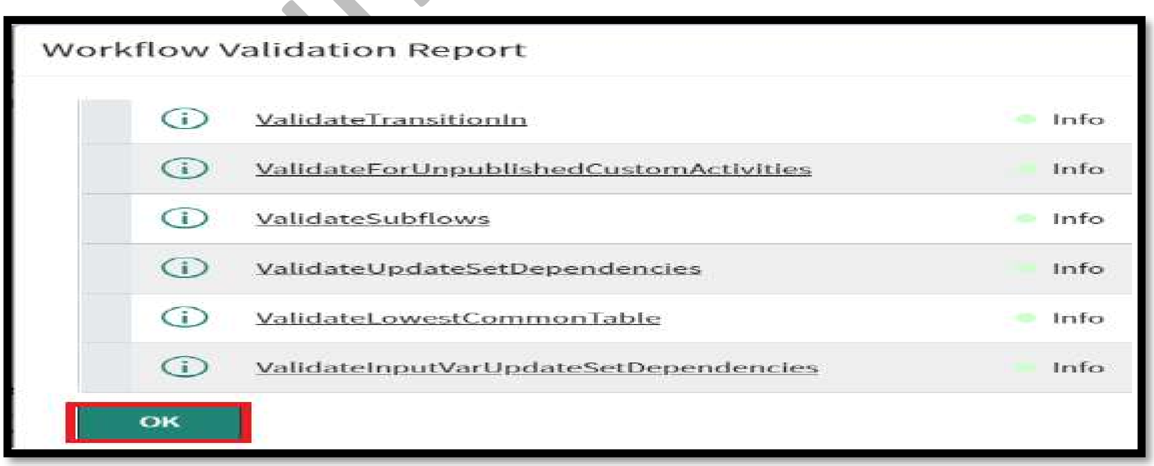
After adding all activities to our workflow, then we can publish it to make it available for business requirement.

Procedure

1. Validate our work flow to click on **Validation** button



2. Open our workflow **actions menu** on the left beside the workflow name.



3. Click on **OK**
4. Click Publish to publish the workflow.

Chapter – 18

Notifications

What is Notification

Notifications will share users informed of events that concern them. The system can notify users by email, SMS text message, or push notification.

Use email notifications to send selected users email or SMS notifications about specific activities in the system, such as updates to incidents or change requests.

If we want to change how the instance processes incoming email, see Inbound email actions.

Note: only administrator can configure email notifications

1. When to send the notification
2. Who receives the notification
3. What content is in the notification

Email Properties

1. All email notifications use the email properties that you define and the email accounts that you set up.
2. Email administrators set up **email accounts** to allow the system to connect to external mail services such as **POP3, SMTP, or IMAP servers**. You can use the Service Now-provided email accounts or create your own accounts for your own email services.
3. Use **Email Properties** to configure settings for inbound and outbound email. Email properties apply to all email accounts.
4. For instances that use a **standard email configuration**, the system cannot send or receive emails that are larger than **25MB**, including the email **header, body text, and attachments.** Instances configured to use an **alternate email configuration** may support different maximum total file sizes. The maximum email size limit is enforced regardless of any configured attachment size limits.

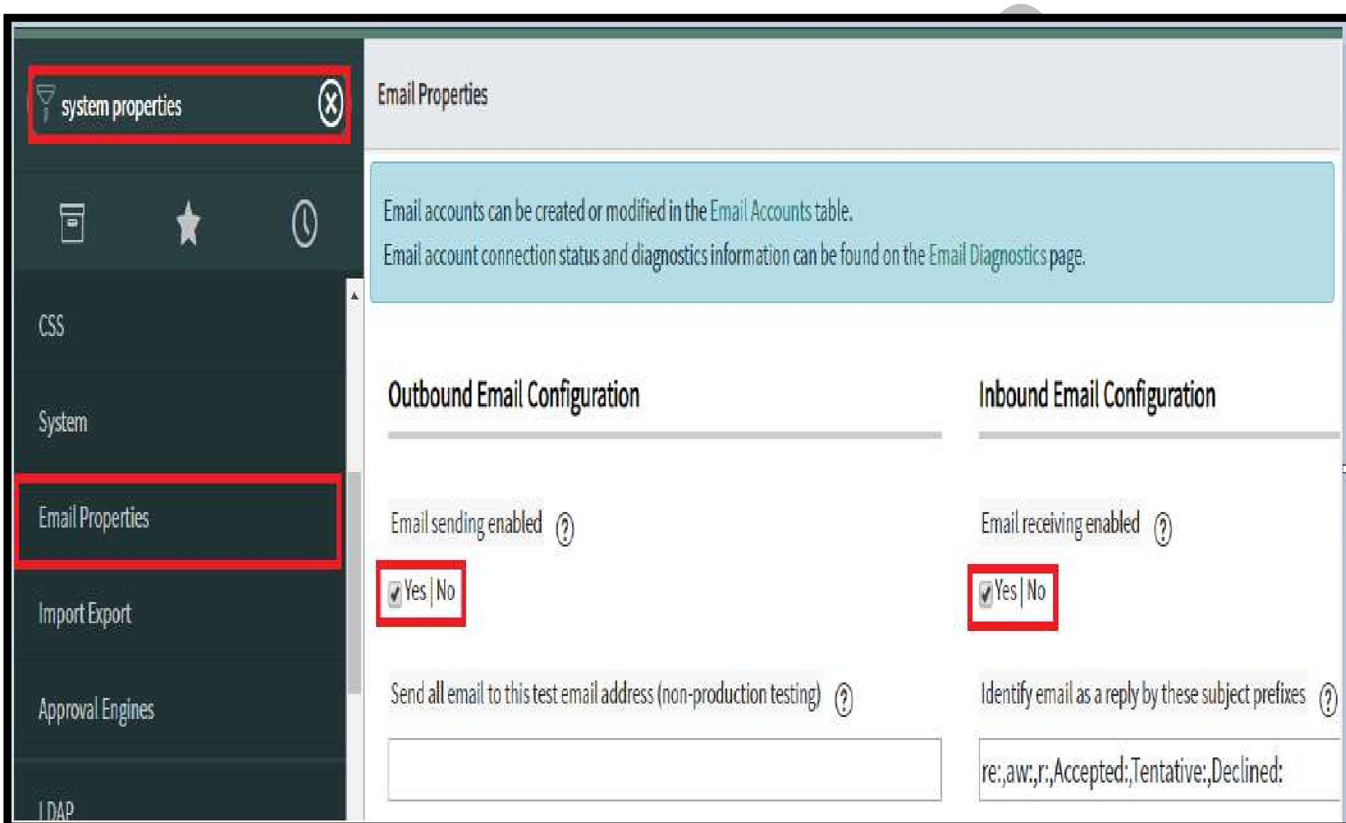
Even if the email and its attachments are less than **25MB**, the actual size of an email message could be larger when it gets encoded by an email client. So these messages might exceed the limit when they reach the instance.

Exercise 1: Enable Basic Email

If we want send are receiving email notification, first we have to enable service now email accounts

Procedure

1. Navigate to **System Properties > Email Properties**
2. Enable Email **Sending** and **Receiving** checkboxes



3. Click on **Save**

Following conditions are must be evaluating to create or update existing email notification

1. Our notification recipients must be **Active Users** and have a valid email address defined.
2. Service Now users or members of groups must be defined as active users in the User **[sys_user]** table.
3. They must also have a valid email address defined for their primary channel (device) in the Notification Device **[cmn_notif_device]** table.
4. If users do not have an active profile and a valid email address, they will not receive notifications.
5. Your notification recipients must have the appropriate notification preferences enabled.
6. If the notification is **subscribable,** each Service Now user or group member must have the notification and channels (devices) for the notification enabled in **their notification preferences**.
7. Administrator can impersonate users to **Review and Configure** their notification preferences.
8. After we create or update the any notification, we can use the **Preview Notification** option to review it.

Following Tabs to use to send Email notifications

1. **When to send —>** Conditions required to send the notification.
2. **Who will receive —>** Recipients of the notification.
3. **What it will contain —>** Contents of the notification.
4. **What Digest will contain —>** Contents of the **email digest** if the notification can be delivered in a digest.

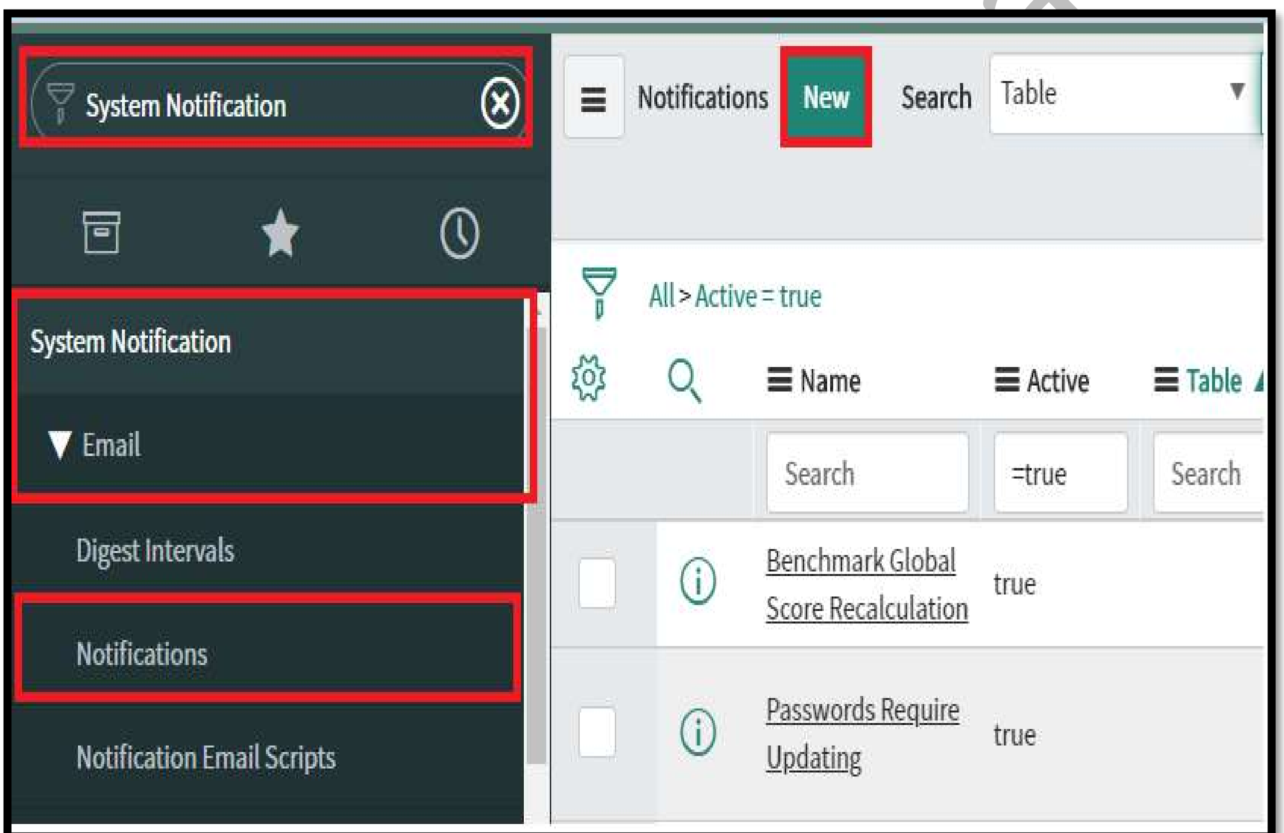
Note: If you do not see all the tabs or fields on the form, switch to the Advanced view then we can be able to see all options

Exercise: 2 How to Create an email notification

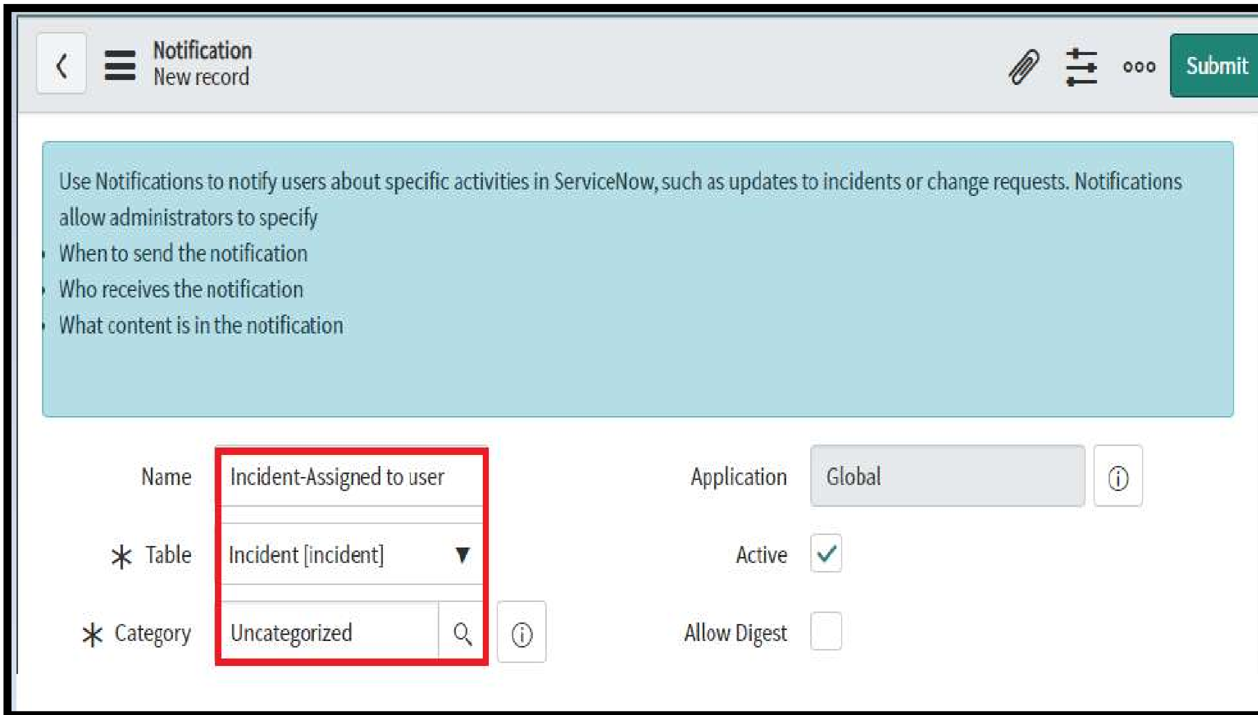
We can create an email notification that will be or specifying when to send it, who will receive it, what it contains the body and subject, and if it can be send in an email digest.

Procedure

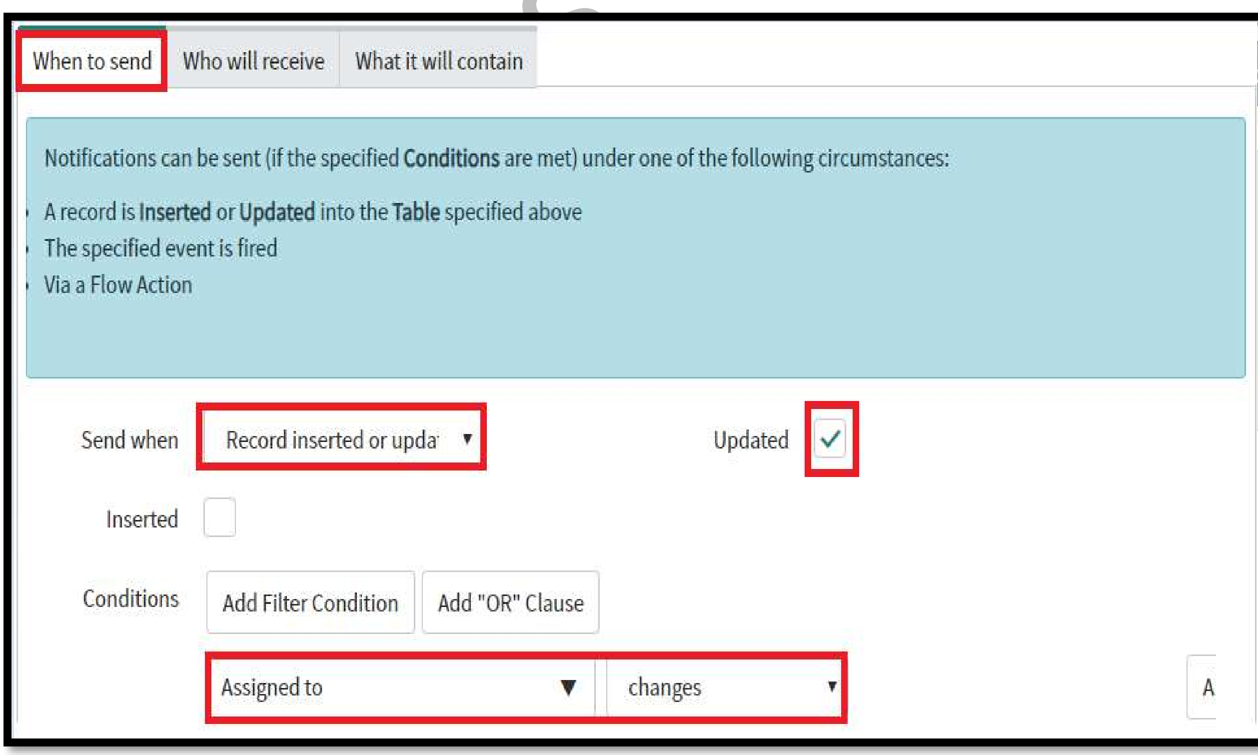
1. Navigate to **System Notification > Email > Notifications.**
2. Click **New.**



3. Fill **Notification** form

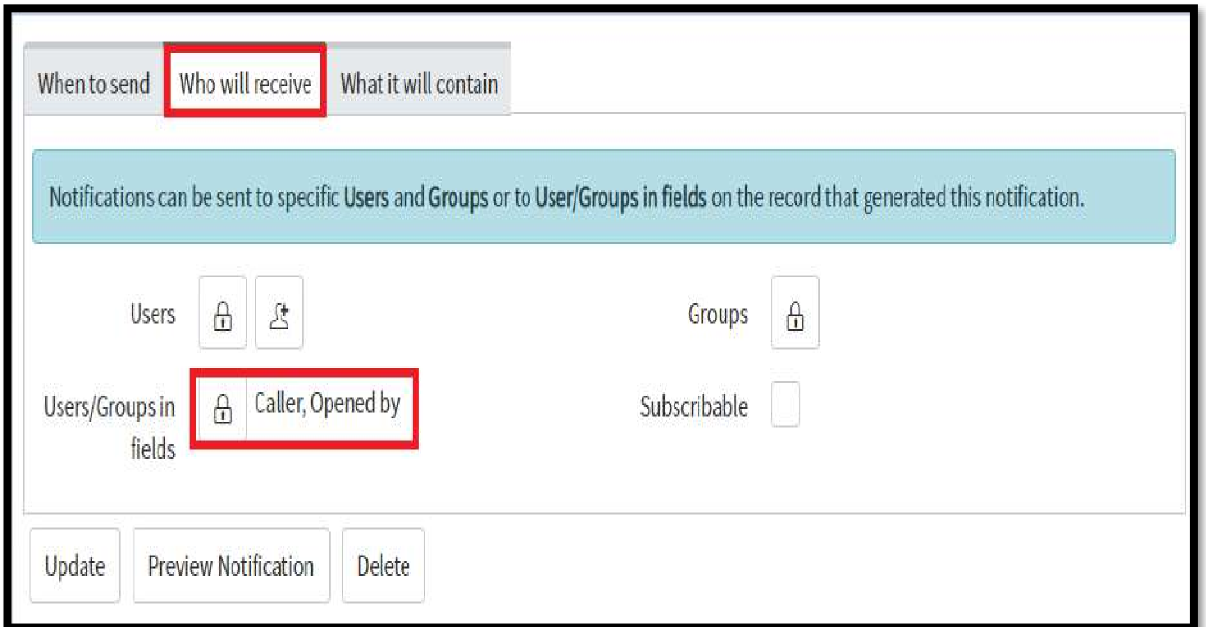


4. **Name:** Incident – Assigned to user
5. **Table:** Incident
6. **Category:** Uncategorized

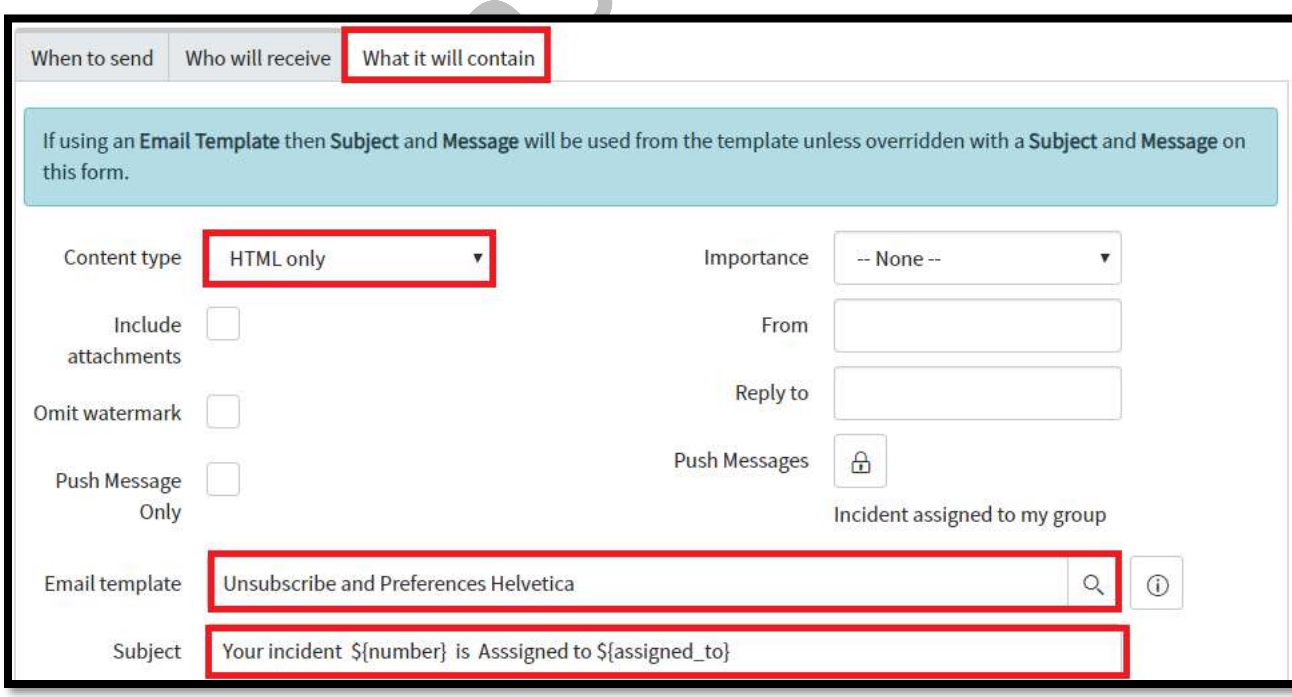


7. Open **When to send** Tab

8. **Send When:** Record **Insert or Updated**
9. **Updated:** Checked **True**
10. **Conditions: [Assigned to] [changes]**



11. Open **Who will receive** Tab
12. **Users/Group:** Caller, Opened by



13. Open **What it will contain** Tab

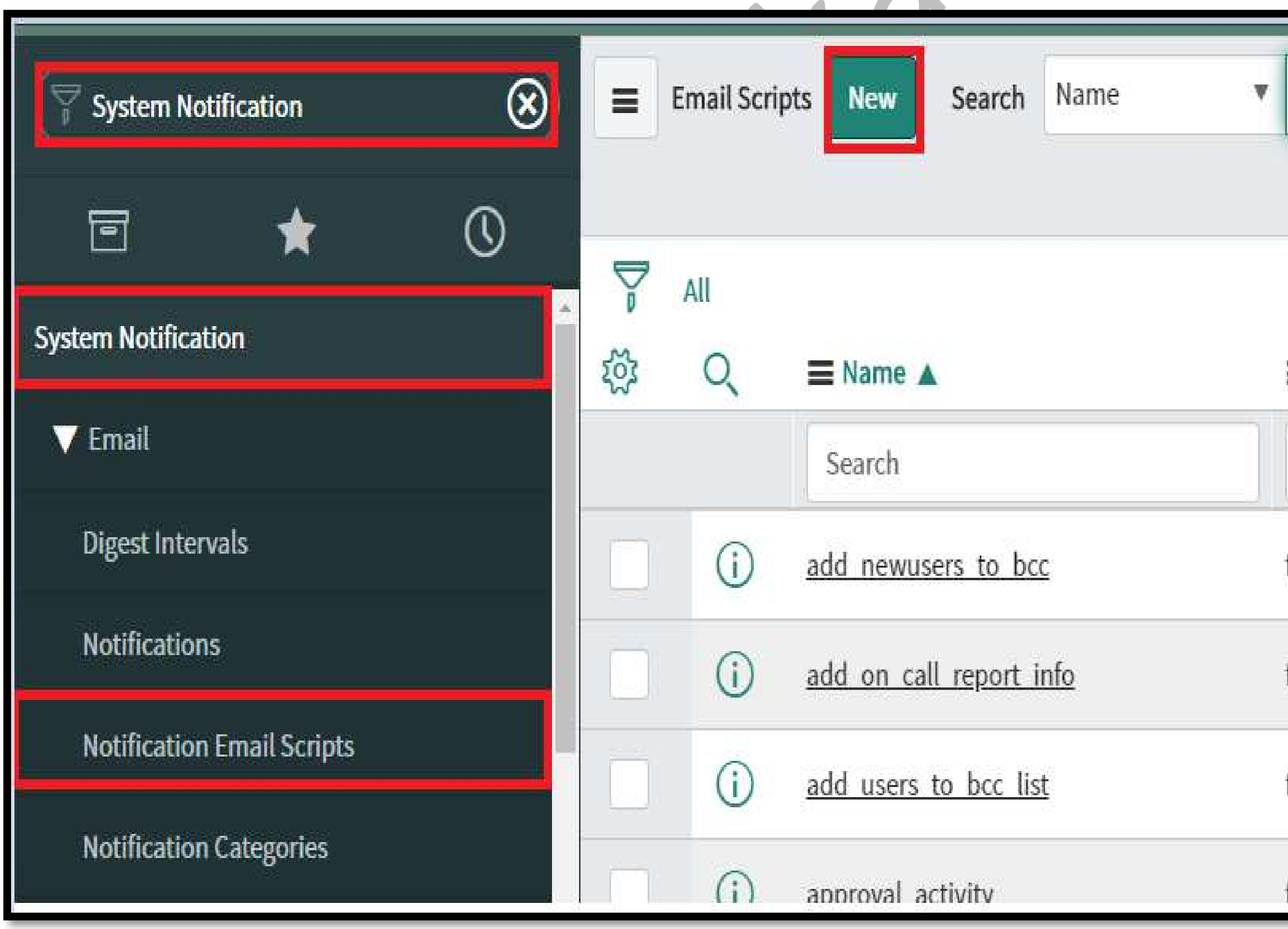
14. Content Type: HTML only
15. Email Template: Unsubscribe and Preferences Helvetica
16. Subject: Your incident **${number}** is Assigned to **${assigned_to}**

Exercise: 3 Create Notification Email Scripts

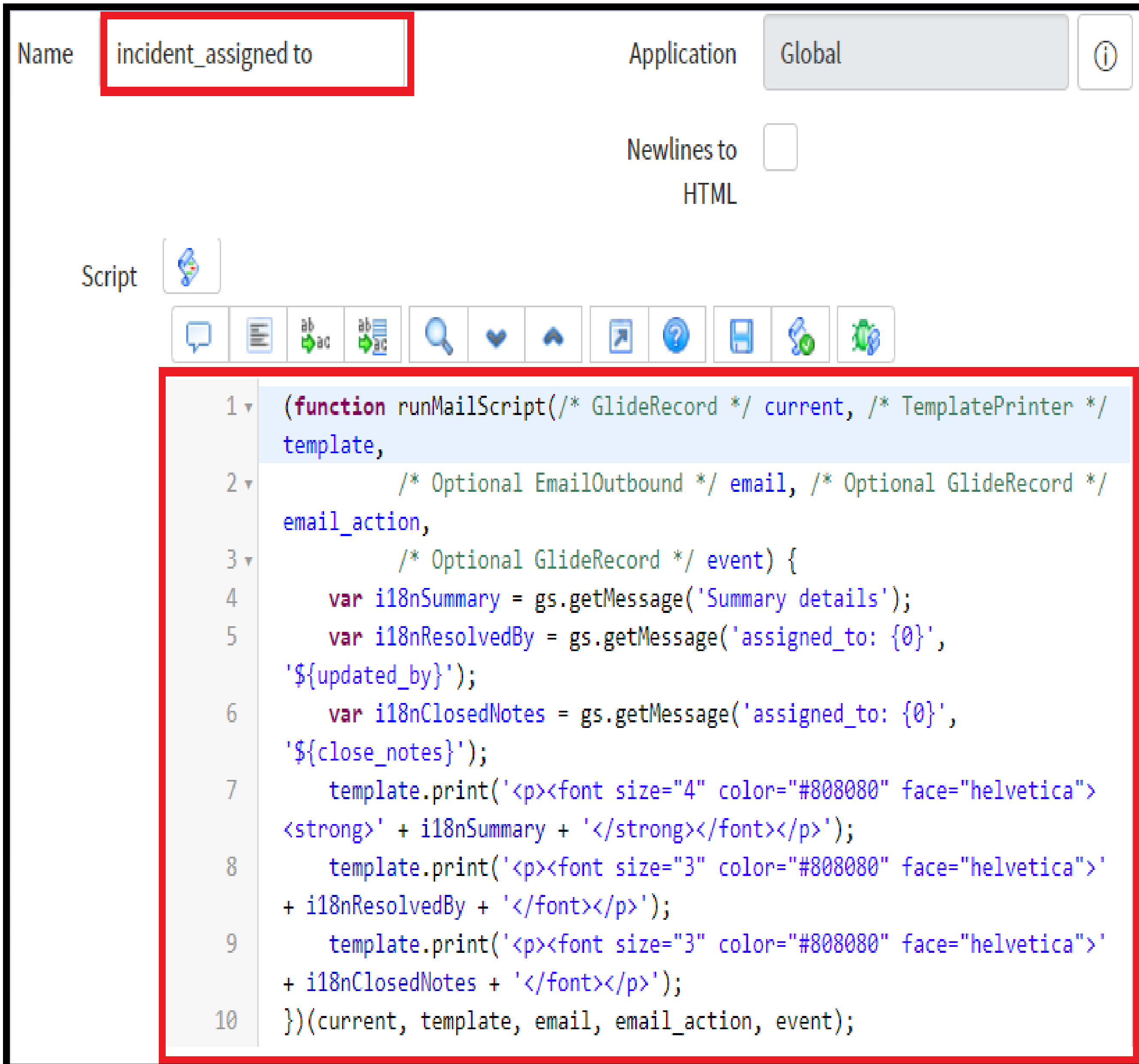
We are using **Notification Email Scripts** to print or send Notification message content from a server-side script. Although Notification Email Scripts are application files, we would create them in the main Service Now browser window.

Procedure

1. Navigate to **System Notification > Email > Notification Email Scripts**
2. Click on **New**



3. Fill **notification email script** form



4. **Name:** incident_assignedto

5. Script :

```
(function runMailScript(/* GlideRecord */ current, /* TemplatePrinter */
template,
          /* Optional EmailOutbound */ email, /* Optional GlideRecord */
email_action,
          /* Optional GlideRecord */ event) {
    var i18nSummary = gs.getMessage('Summary details');
    var i18nResolvedBy = gs.getMessage('assigned_to: {0}',
'${updated_by}');
    var i18nClosedNotes = gs.getMessage('assigned_to: {0}',
'${close_notes}');
    template.print('<p><font size="4" color="#808080" face="helvetica">
<strong>' + i18nSummary + '</strong></font></p>');
    template.print('<p><font size="3" color="#808080" face="helvetica">'
+ i18nResolvedBy + '</font></p>');
    template.print('<p><font size="3" color="#808080" face="helvetica">'
+ i18nClosedNotes + '</font></p>');
})(current, template, email, email_action, event);
```

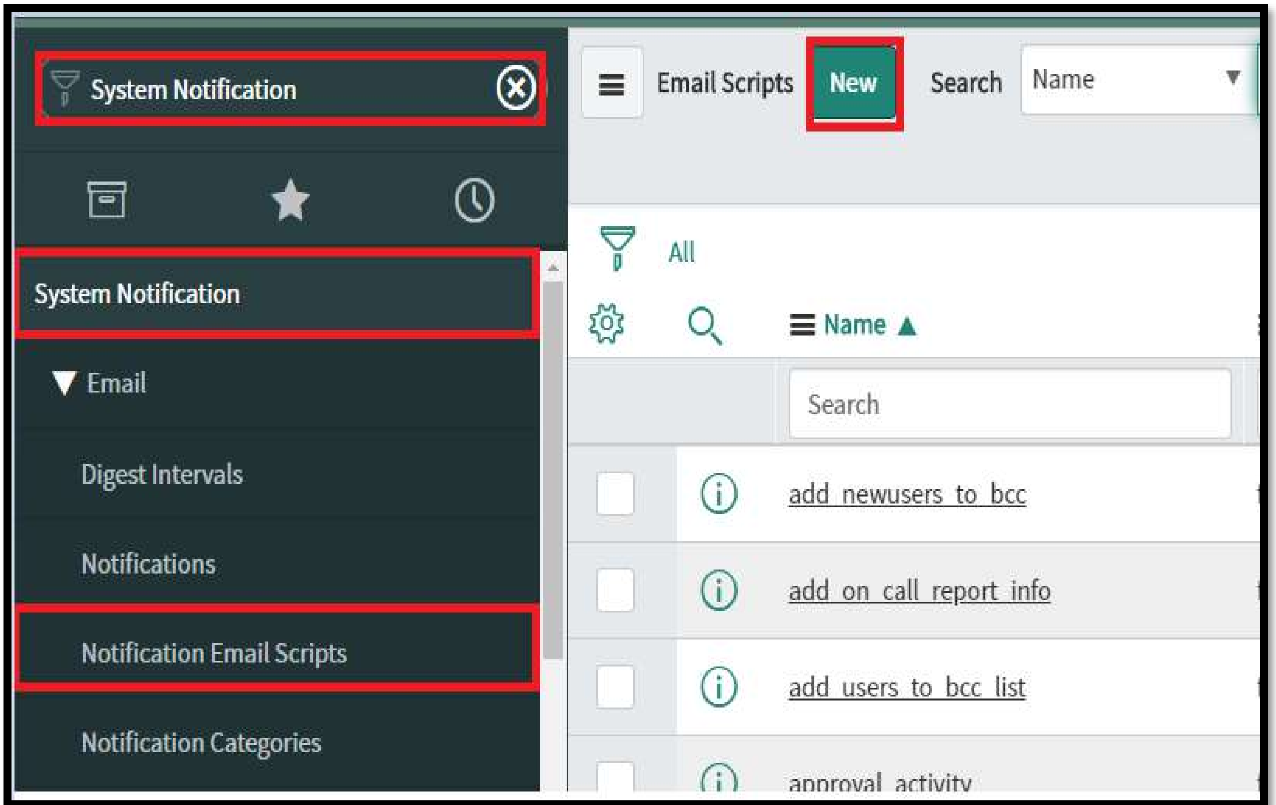
6. Click on **Submit**

Exercise: 4 Crete another notification email script

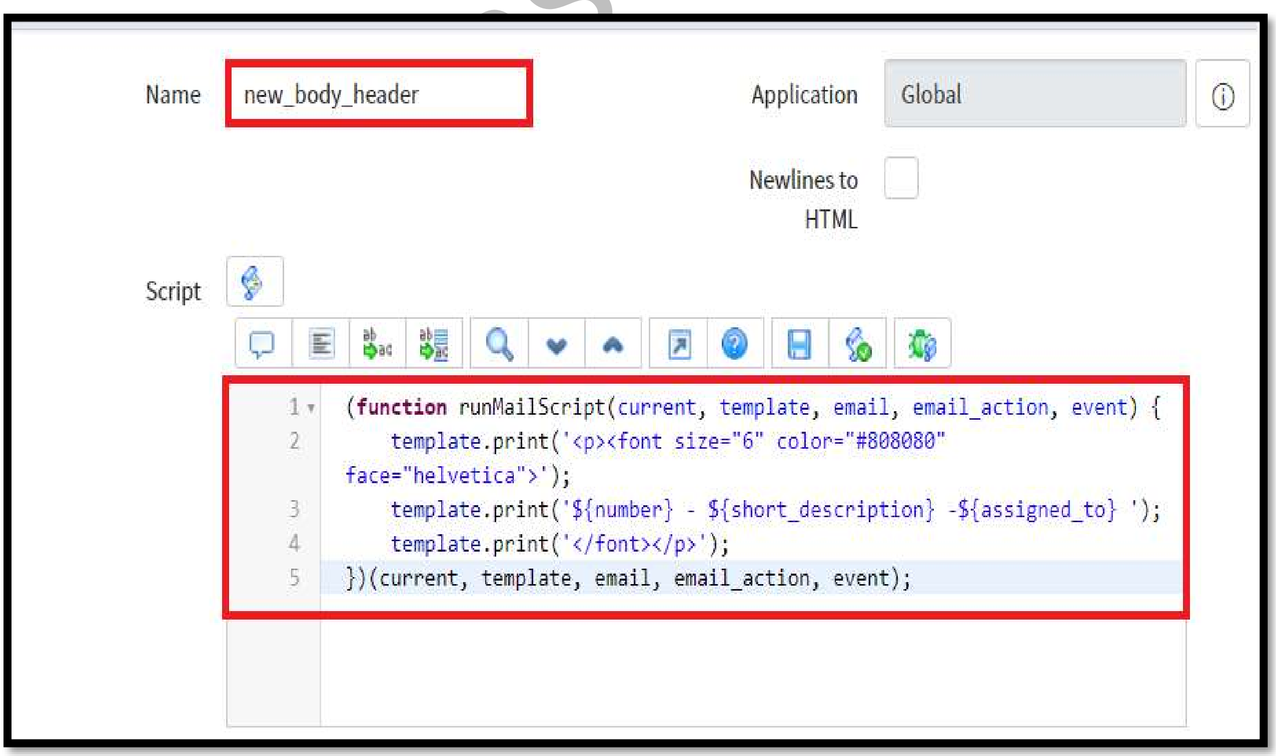
Create one more notification mail script

Procedure

1. Navigate to **System Notification > Email > Notification Email Scripts**
2. Click on **New**



3. Fill **notification email script** form



4. **Name:** new_body_header
5. **Script**

```
(function runMailScript(current, template, email, email_action, event) {
    template.print('<p><font size="6" color="#808080"
face="helvetica">');
    template.print('${number} - ${short_description} -${assigned_to} ');
    template.print('</font></p>');
})(current, template, email, email_action, event);
```

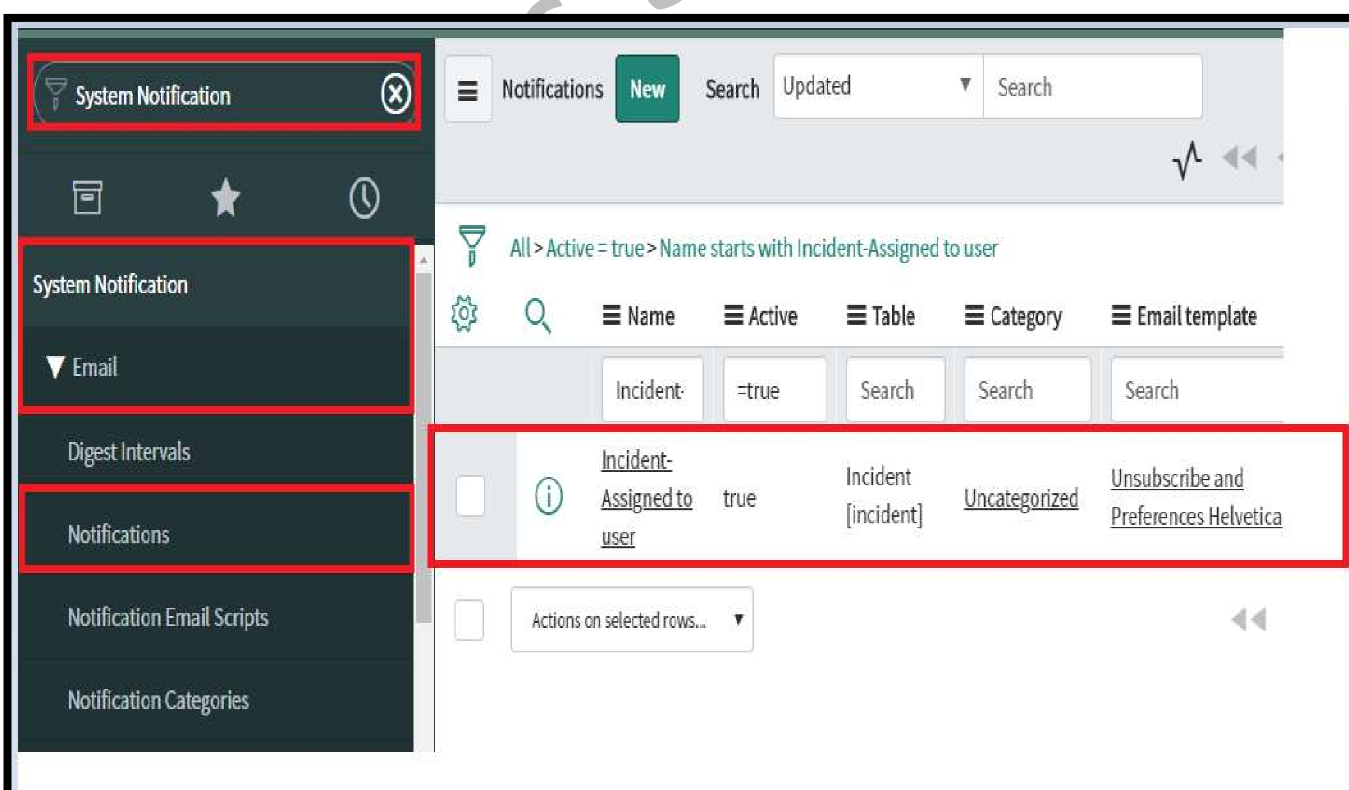
6. Click on **Submit**

Exercise: 5 Add email script to our Notification

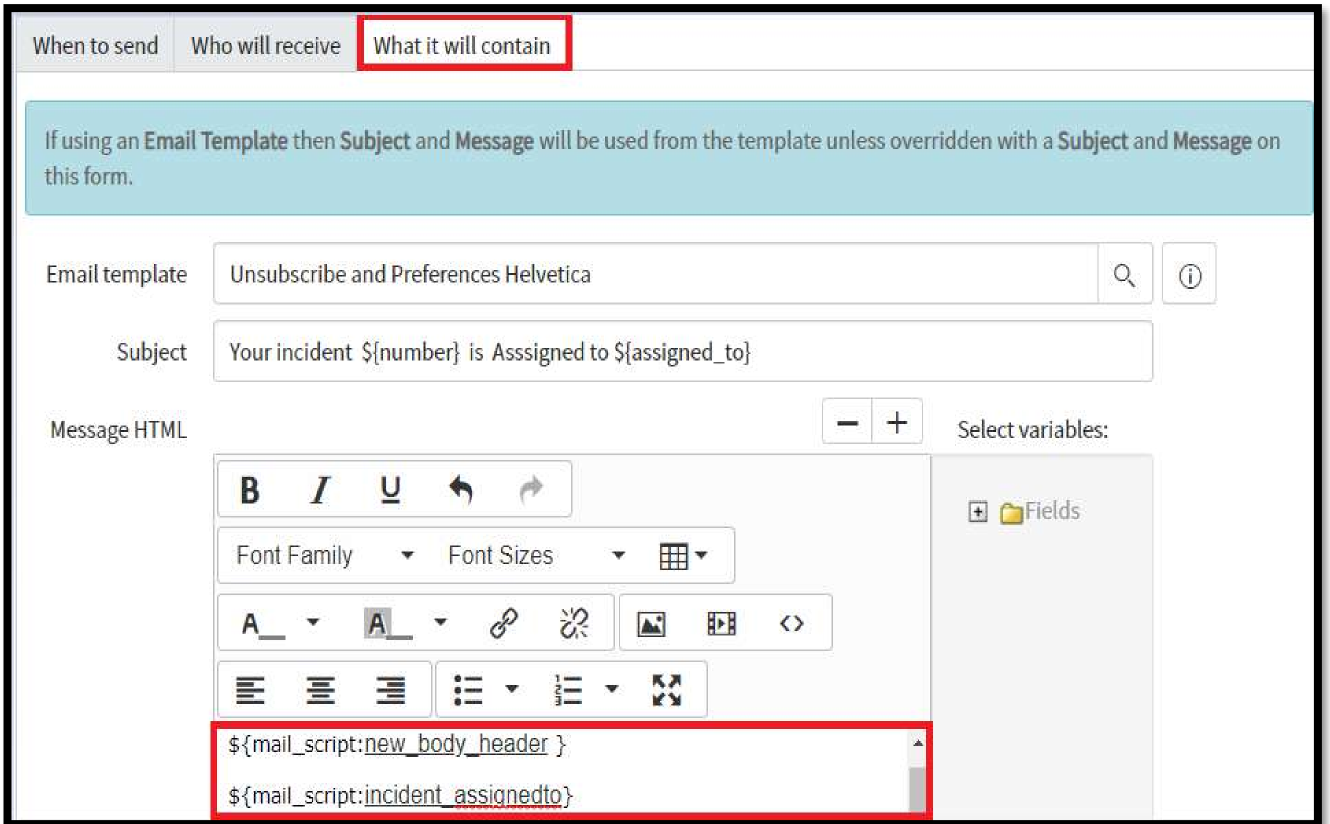
We have to add notification email script to our Notification

Procedure

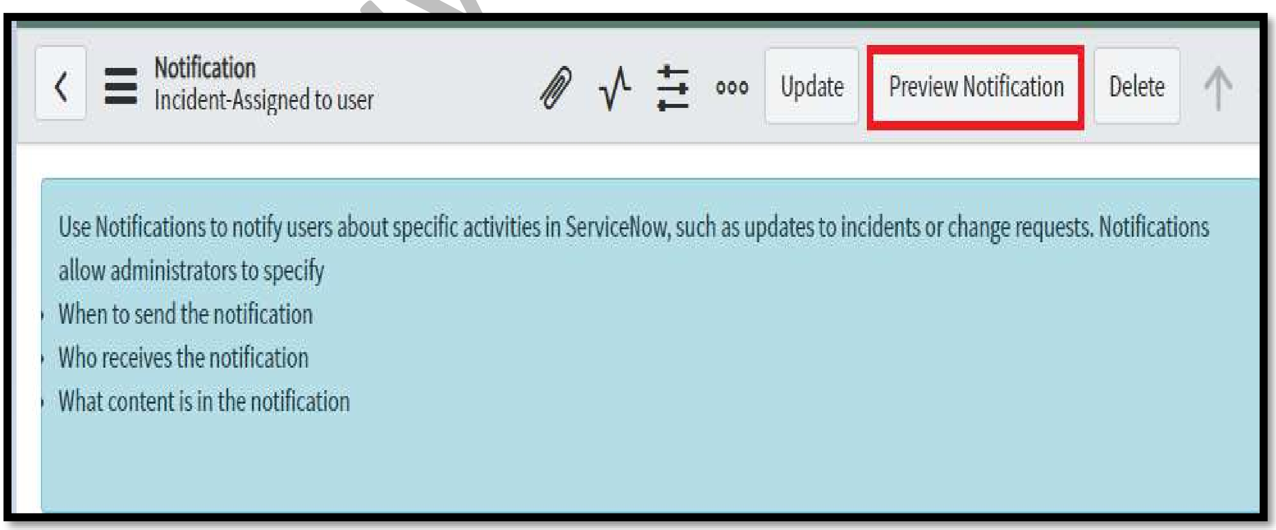
1. Navigate to **System Notification >Email >Notification**
2. Open existing notification record



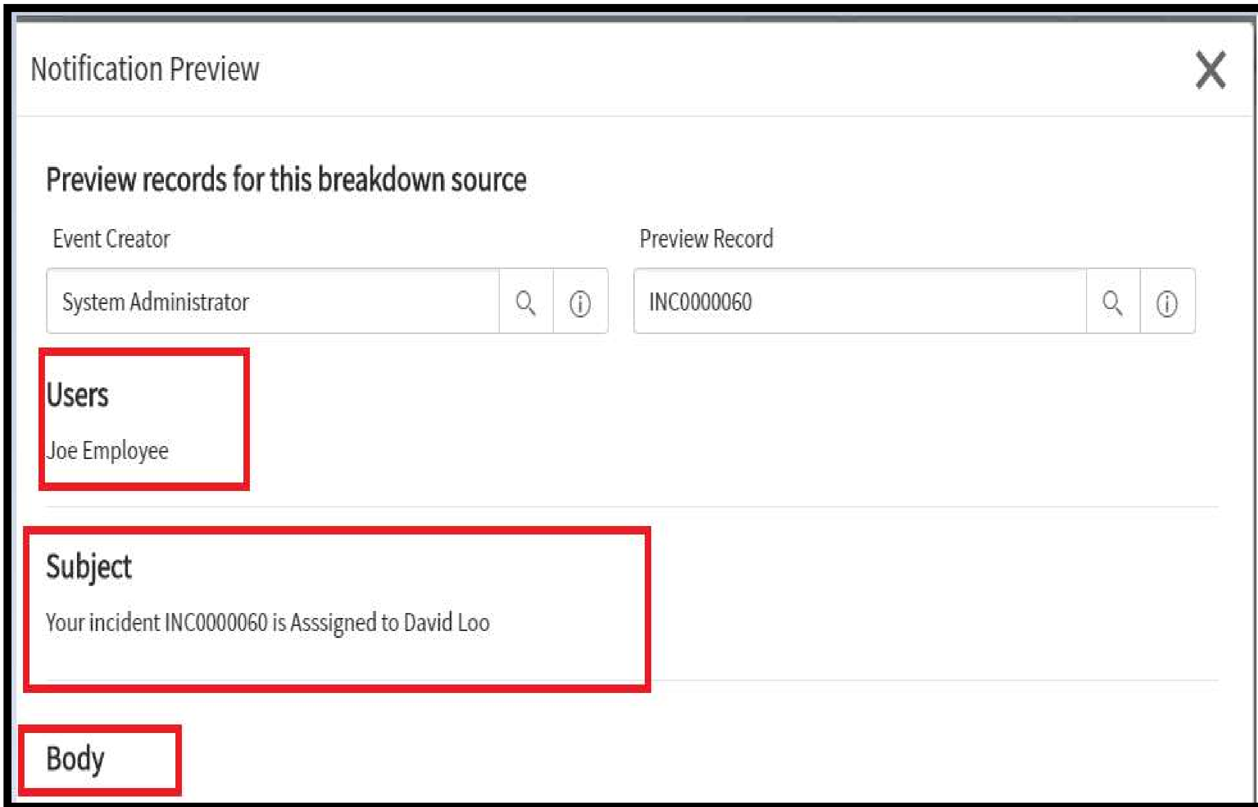
3. Open **What it will contain** tab
4. Add our script into **Message HTML**



5. Add our email scripts like above screen shot



6. Click on **Preview Notification**



7. Check out put for **Notification**

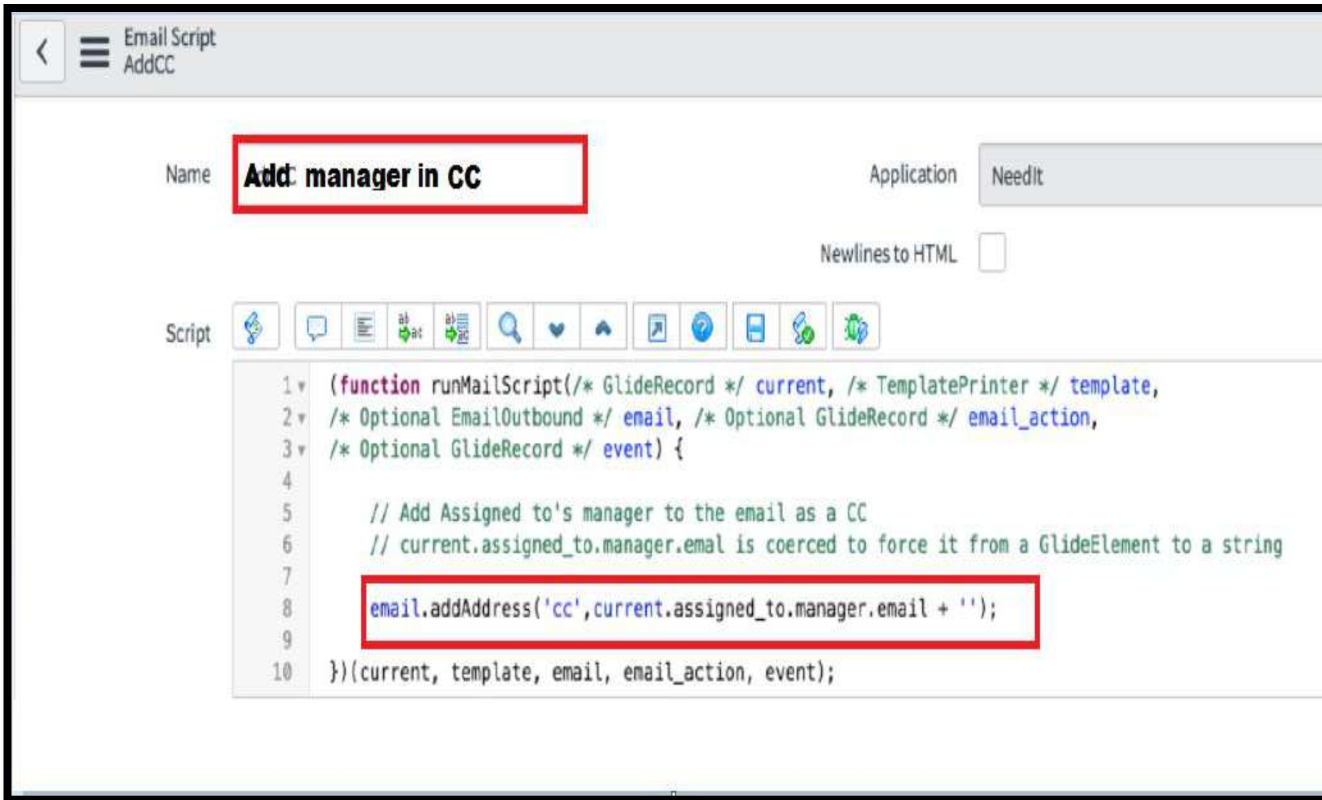
In which condition the notification is sent:

1. When a record is **inserted or updated**
2. When a particular **event is fired**
3. When triggered as an action step in **Flow Designer**

How to send a notification CC and BCC?

If we want send notification **CC and BCC** it's not possible by default, you may have noticed that the notifications Who will receive section does not have fields **for CC or BCC recipients**. Add CCs and BCCs using the email global variable. This example script adds the Assigned to's manager as a **CC on the email.**

1. First we have to Crete an **email script**
2. Then use below code into that



3. When the email is sent to **Beth's manager, Fred Luddy, is a CC**

Chapter – 19

Water Marks

Watermarks on notification emails

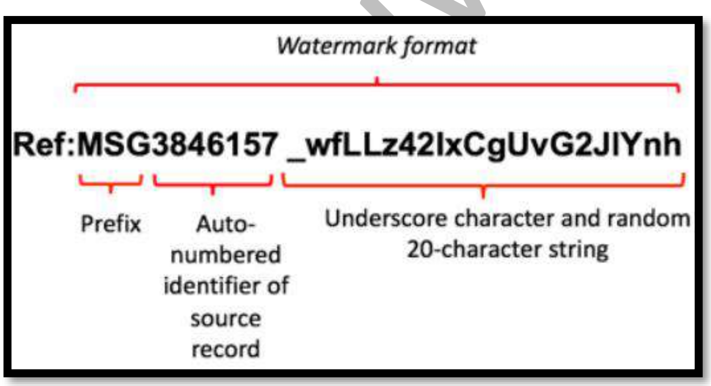
1. By default, the system generates a watermark label at the bottom of each notification email to allow matching incoming email to existing records.
2. Each watermark includes a random 20-character string that makes it unique.
3. The system automatically generates random watermarks for notification emails in base systems.
4. The random 20-character string reduces the possibility of a watermark being guessed or coincidentally matching the watermark of an email from another instance.

Watermark format

The notification email watermark will always begin with **"Ref:"** to identify the label as a watermark. After this identifier, the default label is 31 characters' in length and consists

1. Customizable prefix — The default prefix is **MSG.**
2. Auto-numbered identifier — The numeric string identifying the source record, such as **incident, problem, or change request.**
3. An underscore character followed by a random **20-character** string

Randomized watermark example



When inbound emails are processed, the system matches random watermarks to the appropriate source records

Watermark configuration

Watermarks are always generated, but you can configure them to based on your requirement

1. We can create a custom watermark prefix for each service now instance to prevent accidentally triggering events in the wrong instance.
2. Have custom prefix characters after **MSG**
3. We can have hidden globally
4. We can omitted from individual email messages

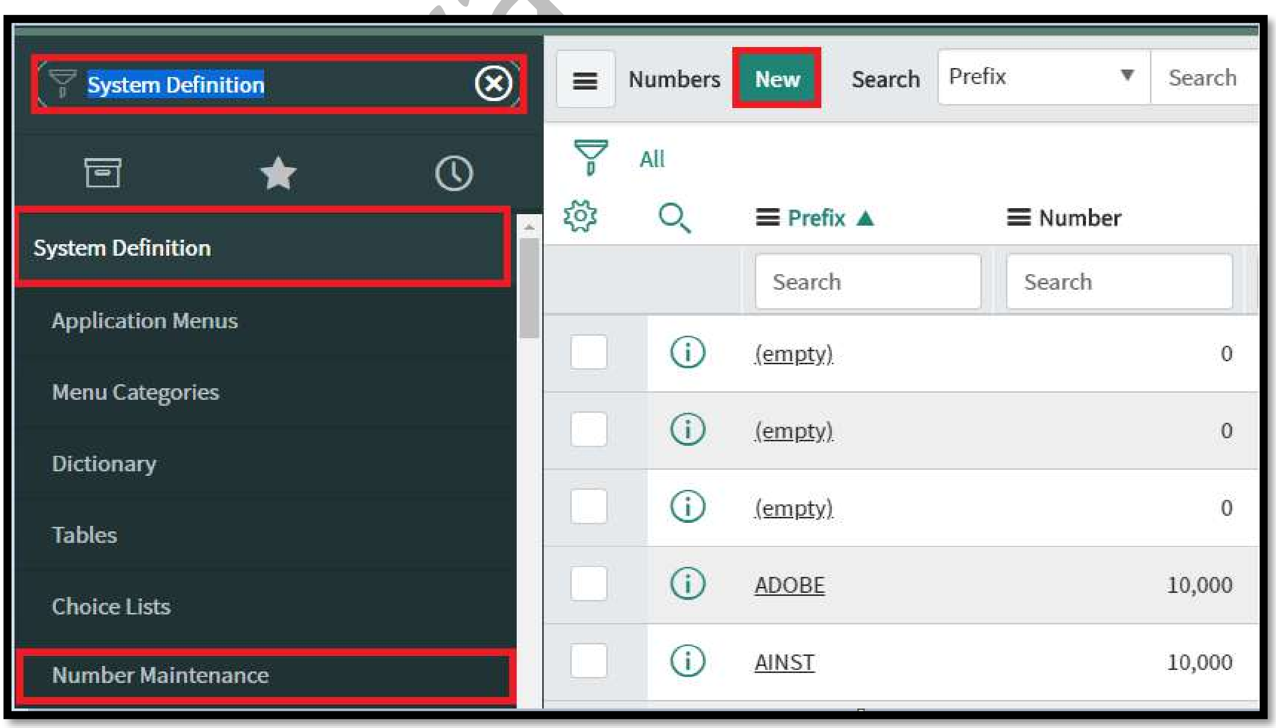
If watermarks are omitted from email notifications, inbound email actions might not work properly. Without a watermark, the system processes inbound email messages as described in Criteria for

Exercise: 1 Create a custom watermark prefix

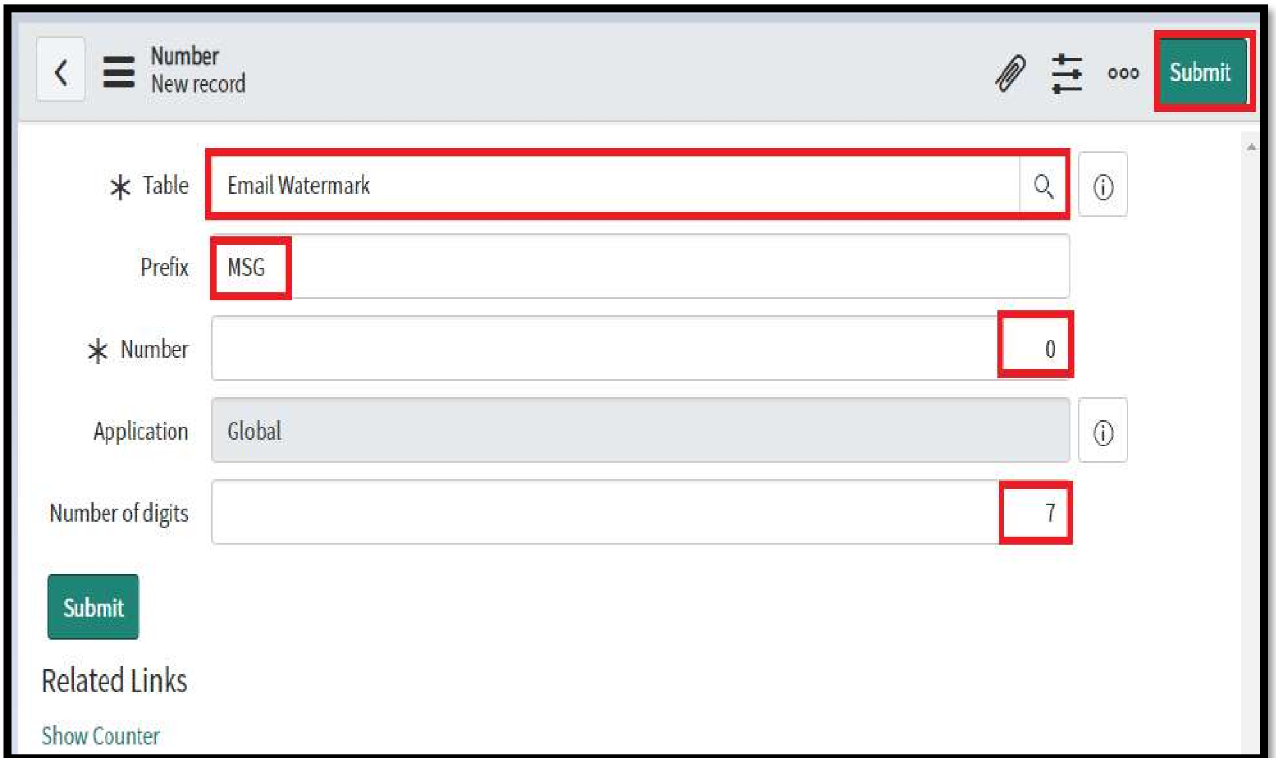
By default, the email notifications use the watermark prefix as **MSG,** but if you want create a custom watermark prefix following below steps.

Procedure

1. Navigate to **System Definition > Number Maintenance.**
2. Click on **New**



3. Fill **watermark** form



4. **Table:** Email Watermark **[sys_watermark]** table.
5. **Prefix:** MSG
6. **Number:** 0
7. **Number of digits:**7
8. Click on **Submit**

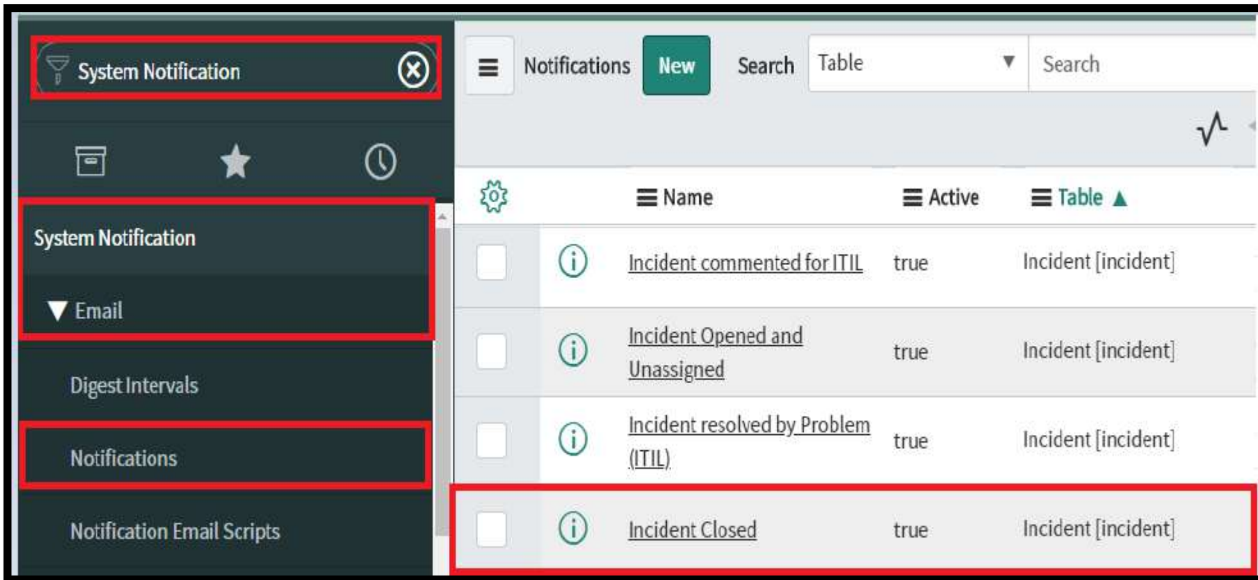
Omit an email notification watermark

We can omit watermarks on email notifications if we do not want the instance to match the notification to an existing record.

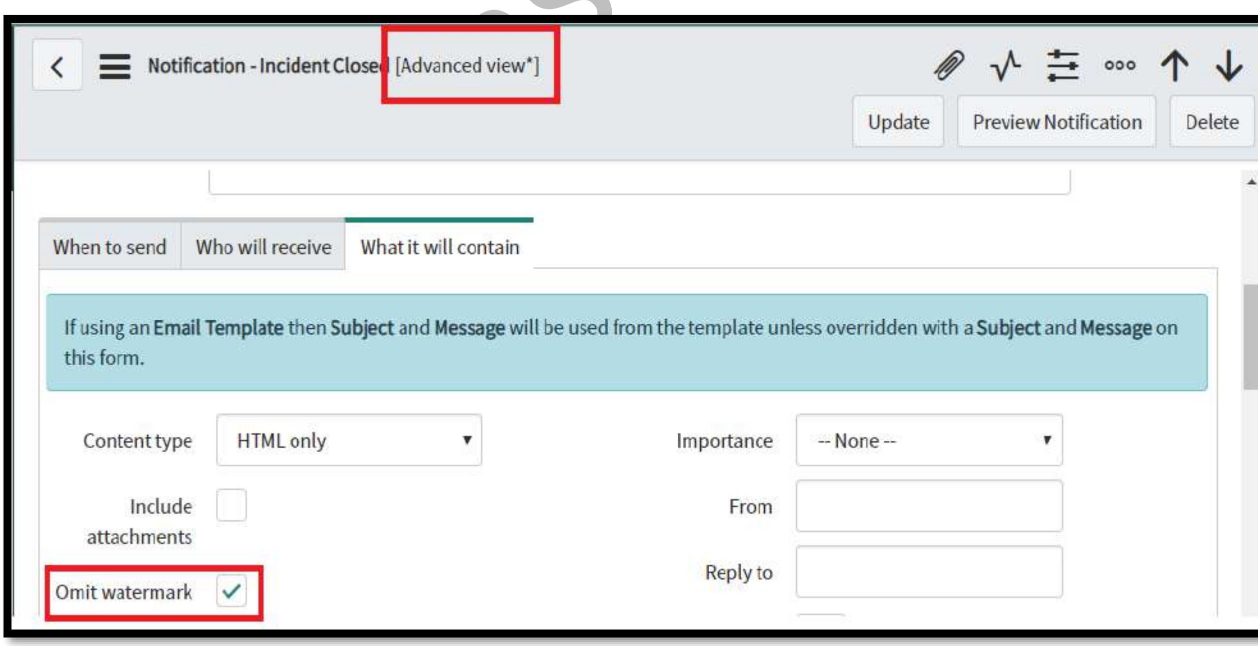
When any incoming emails does not contain a watermark, then the system searches the subject line of mail and message body for an incident number. The system attempts to match any incident number that it finds to an existing incident. If there is a matching incident number, the system updates the incident with the values in the incoming email.

Procedure

1. Navigate to **System Notification > Email > Notifications.**



1. Open selected **email notification** to update
2. Click on **Advanced View** under related link.
3. In the **What it will contain** section, select the **Omit watermark** check box.



4. [Optional] If you want response email messages to generate new incidents, remove the record number **${number}** variable from the Subject and Message **HTML fields.**

Chapter – 20

Access Control List (ACL)

What is ACL (Access Control List)

Access Control List Rules for access control lists (ACLs) restrict the user access to data by requiring users to pass a set of requirements before they can interact with i.e. can be able to restrict every one who can **read, update, create, delete** records into service now tables

1. The applications and modules or objects operation being secured
2. The permissions required to access the applications and modules and objects

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

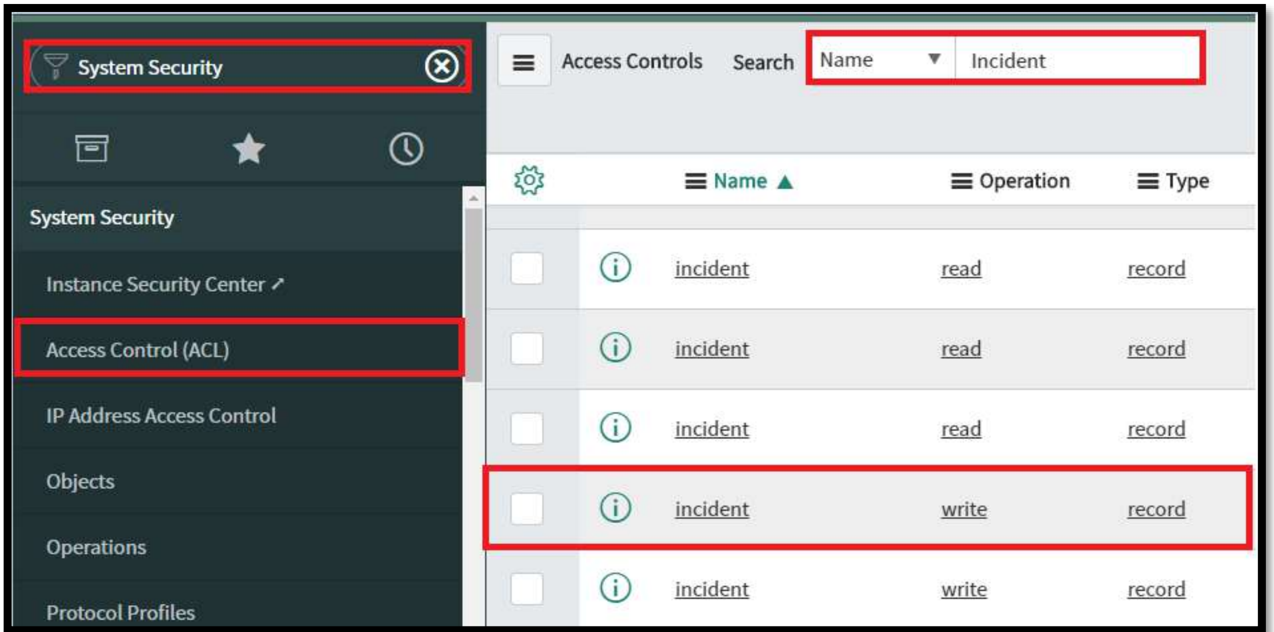
1. The user has one of the roles specified in the Role list, or the list is empty.
2. Conditions in the Condition field evaluate to true, or conditions are empty.
3. The script in the Script field (advanced) evaluates to true, or sets the variable "answer" to true, or is empty.

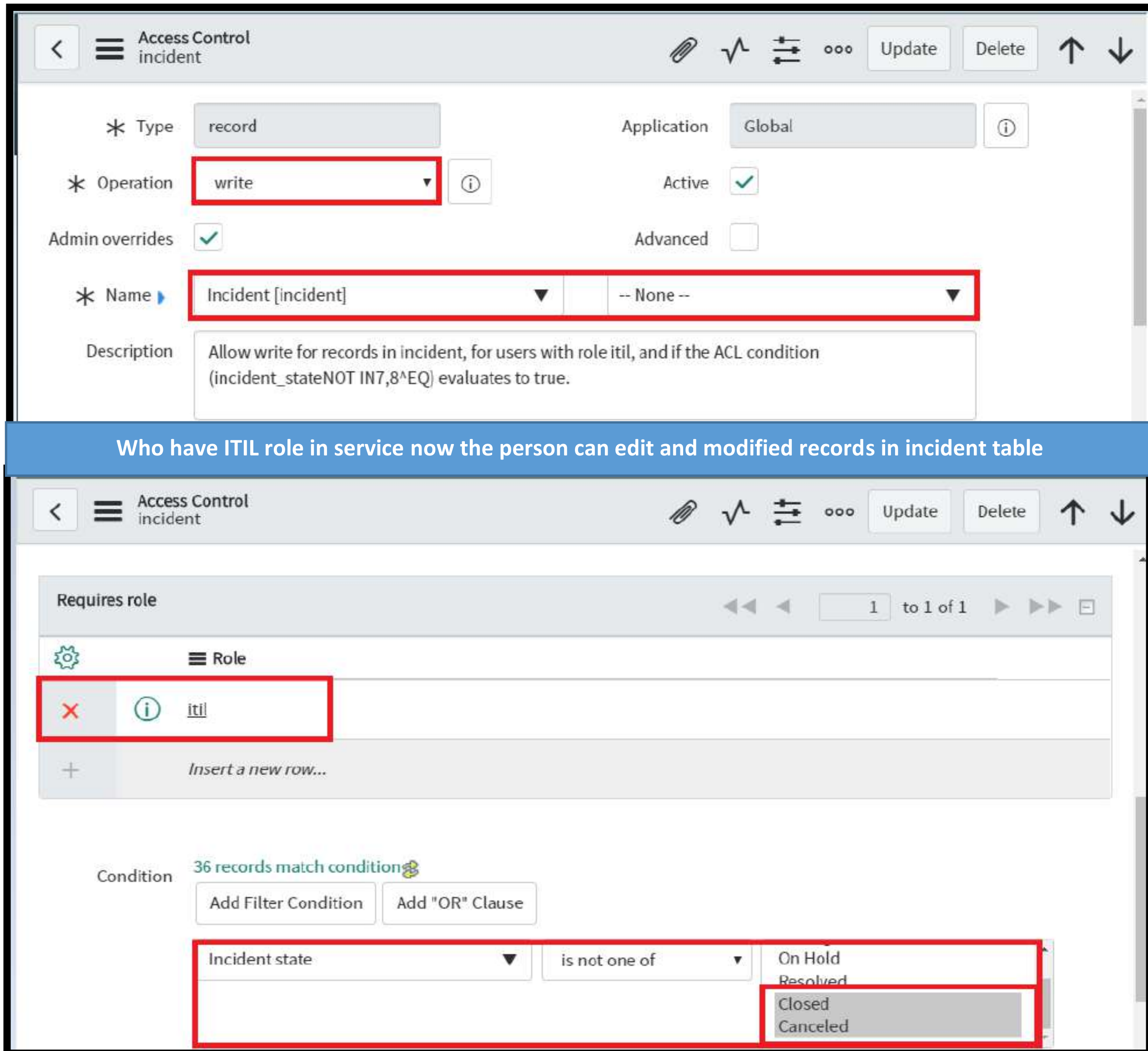
The three checks are evaluated independently in the order displayed above.

To gain access to an object and operation, a user must pass all permissions listed in an access control. For example, this access control restricts access to write operations on the incident table.

Procedure

1. Navigate to **System Security > Access Control(ACL)**
2. **Open Existing ACL write** operation record from incident table


System Security
System Security
Instance Security Center
Access Control (ACL)
IP Address Access Control
Objects
Operations
Protocol Profiles
Access Controls
Search
Name
Incident
Name
Operation
Type
incident read record
incident read record
incident read record
incident write record
incident write record

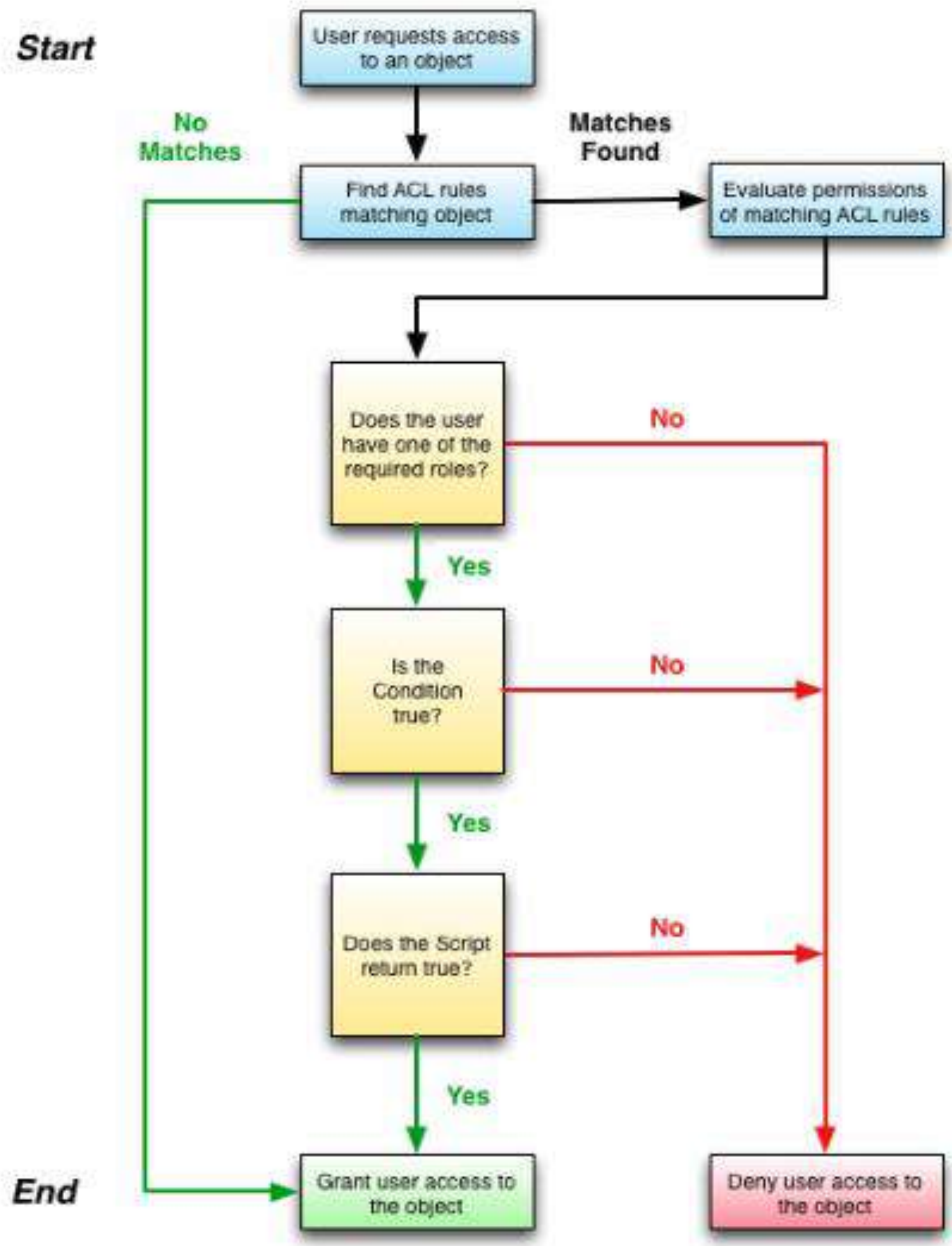


ACL evaluation process

An ACL rule only given a permission a user access to an applications and modules and object if the user meets all of the permissions required by the matching ACL rule and conditions.

1. The **Condition** must evaluate to true.
2. The **script must evaluate to true** or return an answer variable with the value of true.
3. The user must have one of the **roles in the required roles list.** If the list is empty, this condition evaluates to true.
4. [Record ACL rules only] The matching table-level and field-level ACL rules must both evaluate to true.

ACL evaluate permissions



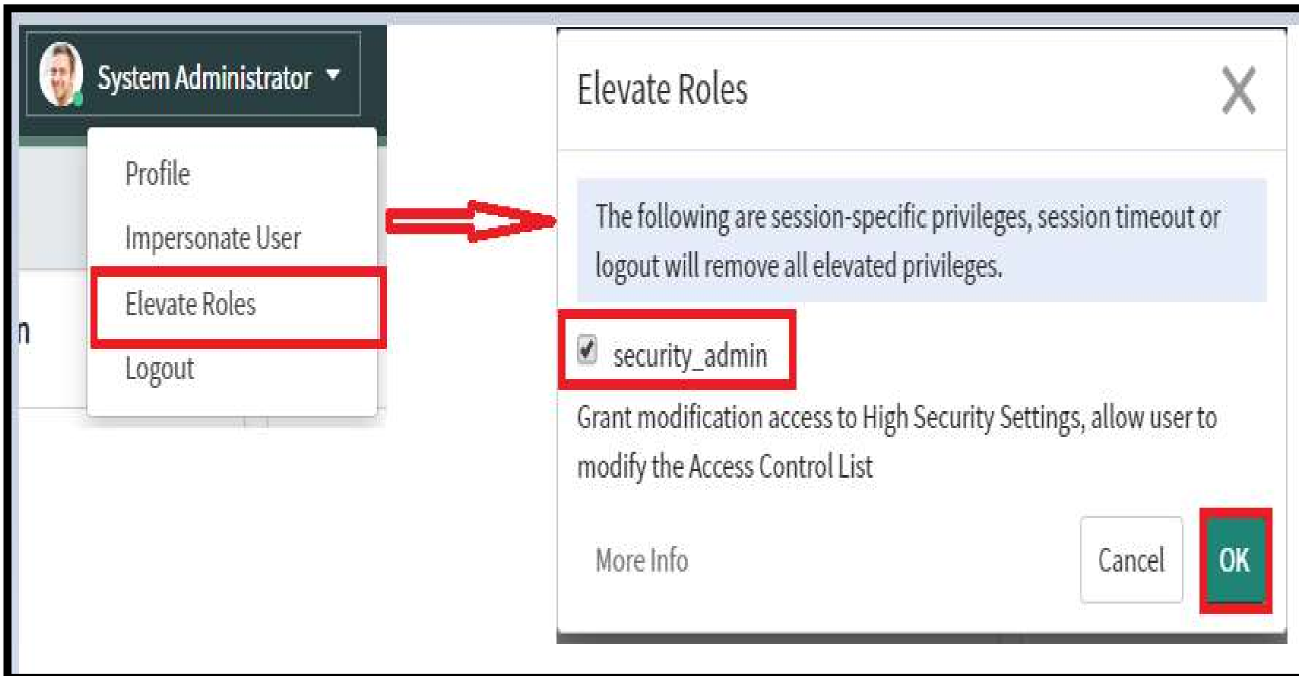
1. Whenever a user requests data, the system can search for access control rules that match the requested object and operation.
2. If there is a matching access control rule, then the system evaluates if the user has the permissions required to access the object and operation.
3. If an access control rule specifies more than one permission, then the user must meet all permissions to gain access to the object and operation.
4. Failing any one permission check prevents the user from accessing the matching object and operation.
5. If a user does not meet the permissions of the first matching rule, the system evaluates the permissions of the next matching access control rule as specified by the access control processing order.
6. If the user fails to meet the permissions of any matching access control rule, the system denies access to the requested object and operation.
7. Depending on the object secured, the ACL rule hides a field on a form, hides rows from a list, or prevents a user from accessing a UI page.

If there are no matching access control rules for the requested object and operation, then the system grants the user access to it. In practice, it is rare for the system to find no matching rules because the system has a set of default access control rules that protect all record operations.

The wildcard field ACL rule **(*.*)** for the create operation reuses the same permissions as the write operation. This means that the create permissions are the same as the write permissions unless you define an explicit create operation ACL rule.

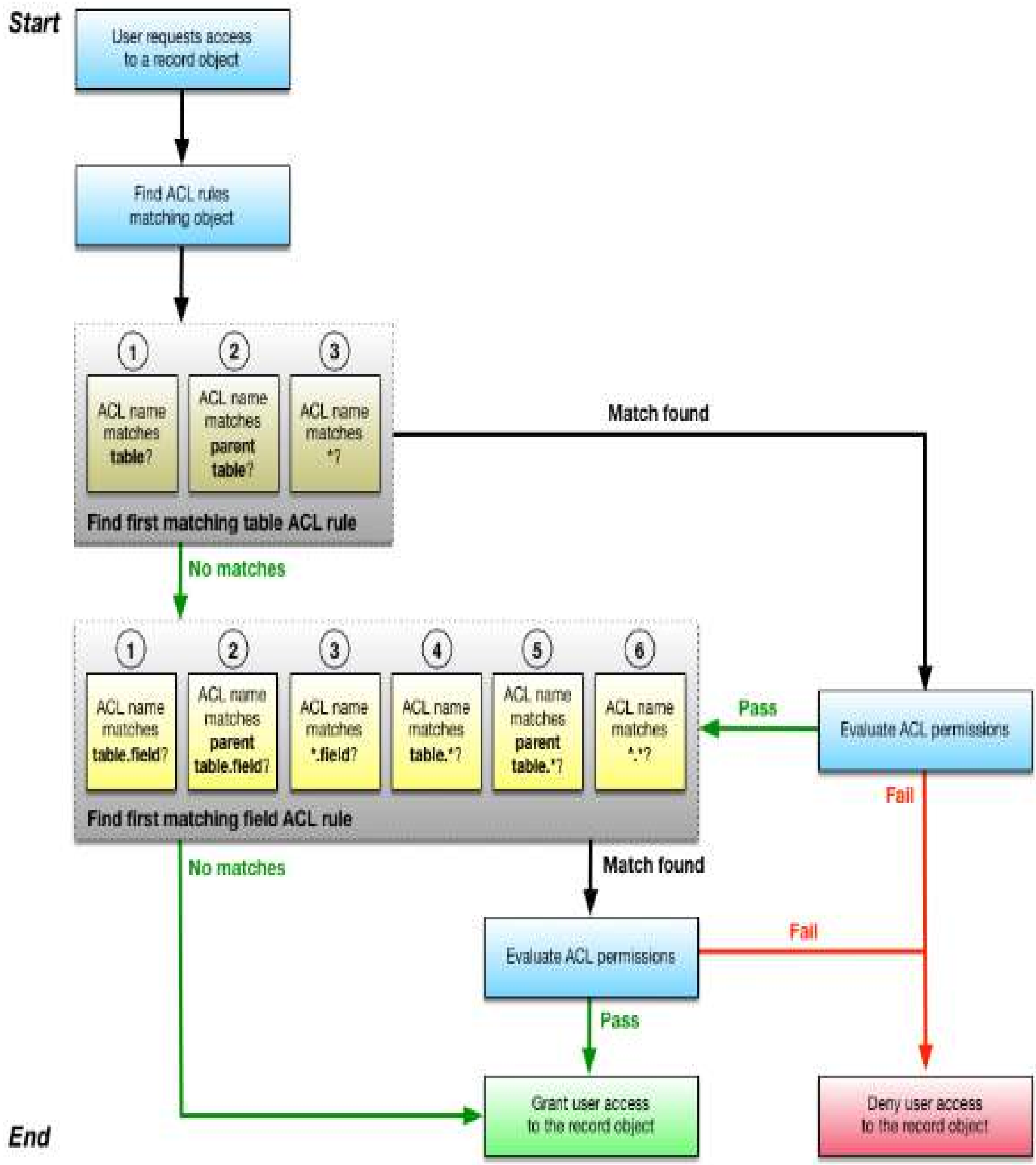
Required role to create ACL Rules

Normal admin users can view and debug access control rules. However, to create or update existing access control rules, administrators must elevate privileges to the **security_admin** role. See **Elevate to a privileged role** for instructions.



1. Expand User menu
2. Click on **Elevate Roles**
3. **Check** security_admin role
4. Click on **OK**

ACL matching Diagram



Levels of ACL Rules

We can implement ACL rules for multiple levels of different component in the instance

1. Table Level
2. Field Level

Table ACL rules

The user must first pass the table ACL rule. Since the base system includes STAR (*) table ACL rules that match every table, the user must always pass at least one table ACL rule. The base system provides additional table ACL rules to control access to specific tables

Table ACL rules are processed in the following order:

1. It must match the **table name**. For example, incident, problem, change
2. It must match the **parent table name.** For example, task, cmdb_ci.
3. It must match any table name **(*).** For example, *.

If a user fails all table ACL rules, the user cannot access the fields in any table. If a user passes a table ACL rule, the system then evaluates the field ACL rules.

Field ACL rules

After a user passes a table ACL rule, field ACL rules are processed in the following order:

1. It must match the table and field name. For example, **incident. caller_id.**
2. It must match the parent table and field name. For example, **task.caller_id**
3. It must match any table (*) and field name. For example, ***. number**
4. It must match the table and any field (*). For example, **incident.*.**
5. It must match the parent table and any field (*). For example, **task.***
6. It must match any table (*) and any field (*). For example, ***.***

A user must pass the table ACL rule to be granted access to the table's fields. For example, the user must first pass the table ACL rule for the incident table to access the **Caller** field in the incident table.

The first successful field ACL evaluation stops ACL rule processing at the field level. When a user passes a field ACL rule, the system stops searching for other matching field ACL rules. For example, if a user passes the field ACL rule for **incident.caller_id**, the system stops searching for other ACL rules that secure the **Caller** field in the incident table.

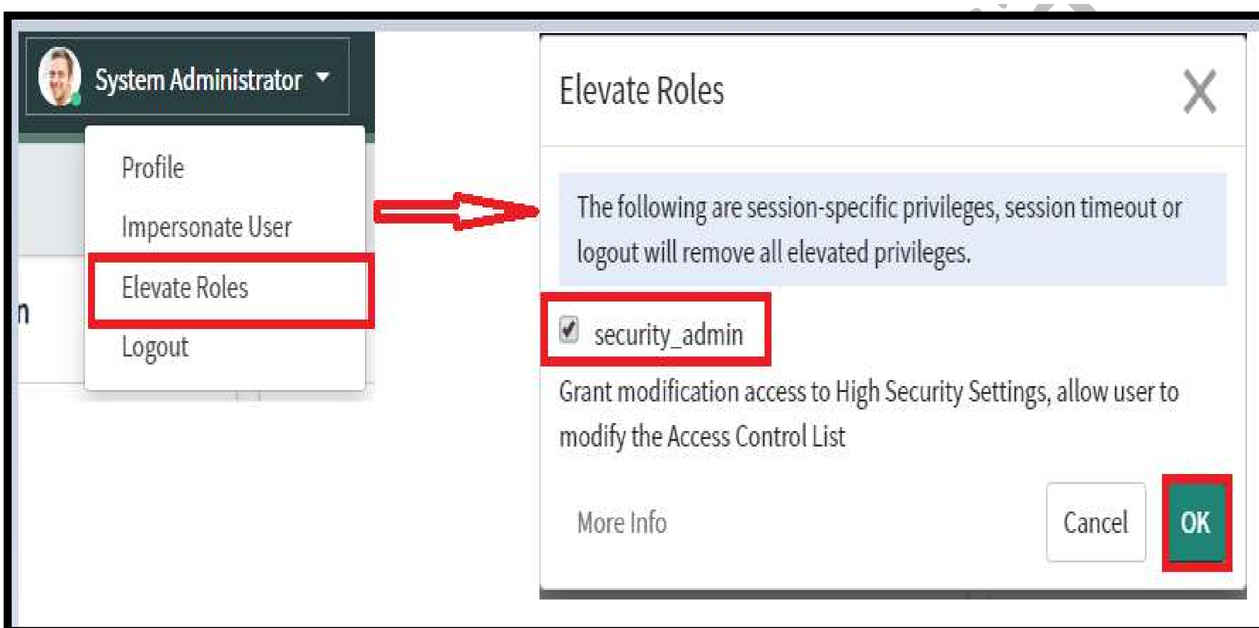
Exercise 1: Create an ACL rule

We can create a custom ACL rule to restrict user access to new objects or to change the default security behavior in our instance.

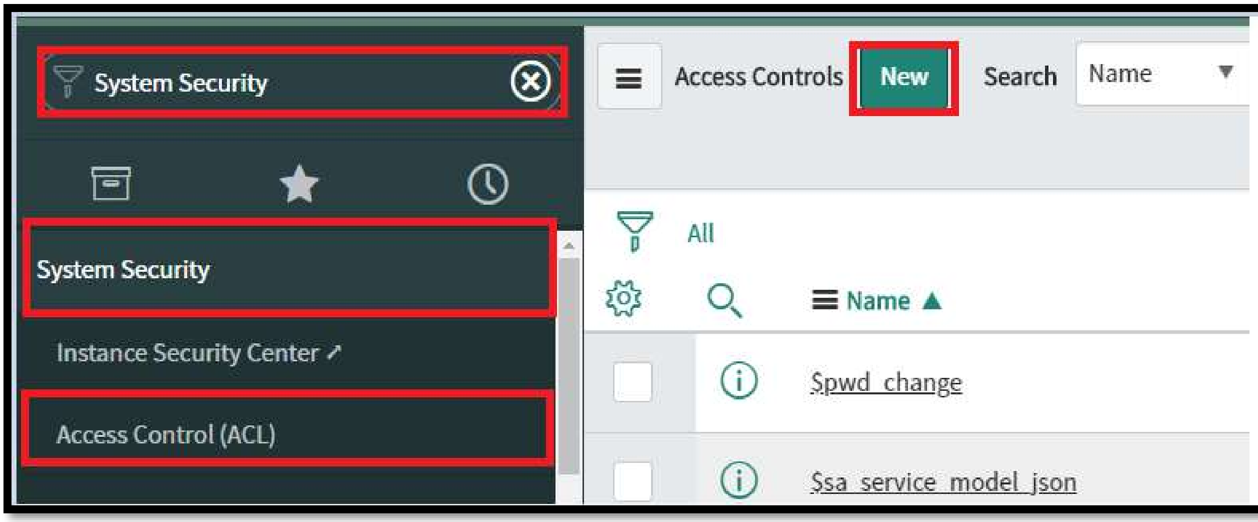
While creating new ACL rules, first you must be elevating privileges to the **security_admin** role. For tables that are in a different scope from the ACL rule record, the types of rules are limited.

Procedure

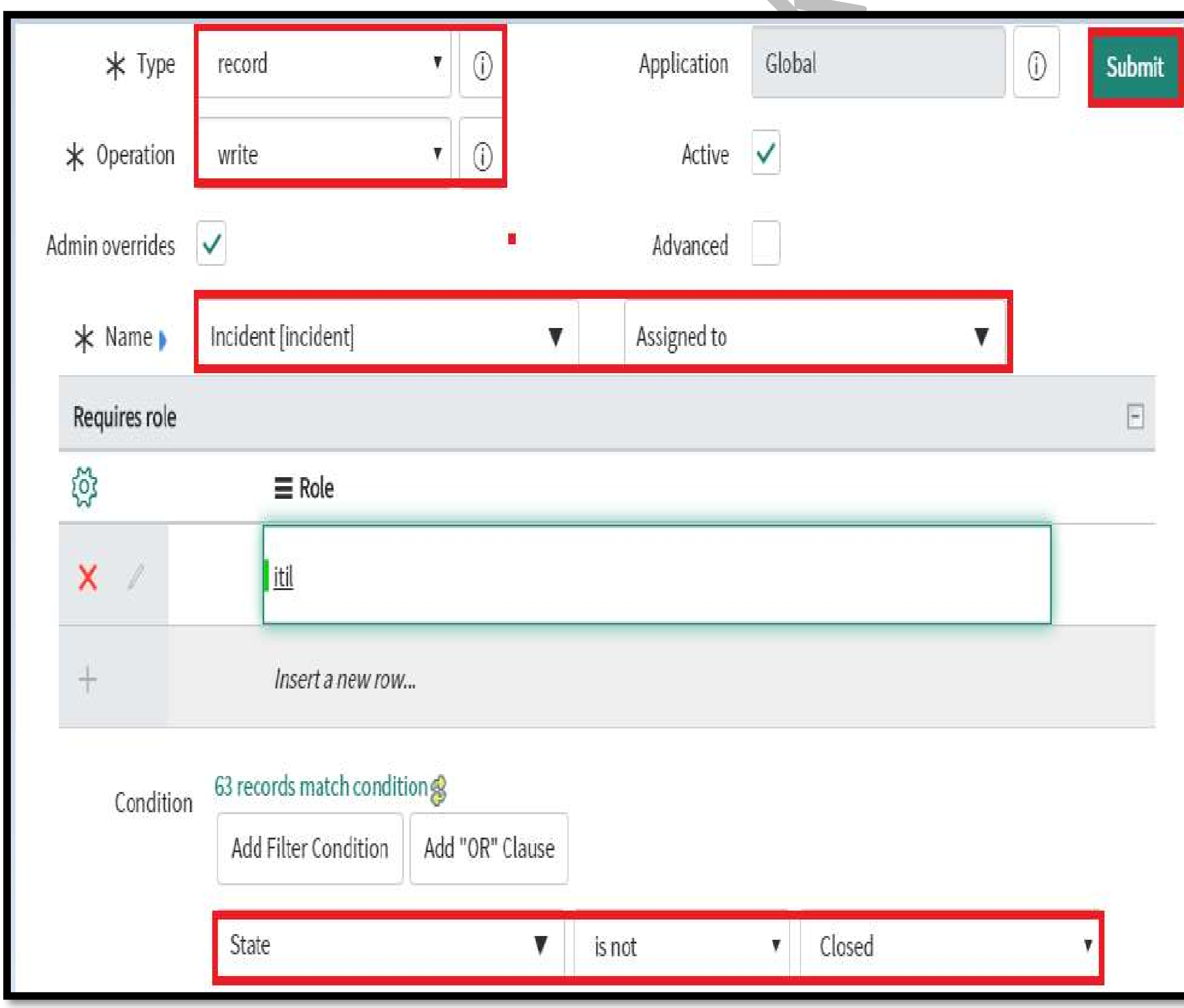
1. **Elevate privileges** to the **security_admin** role.



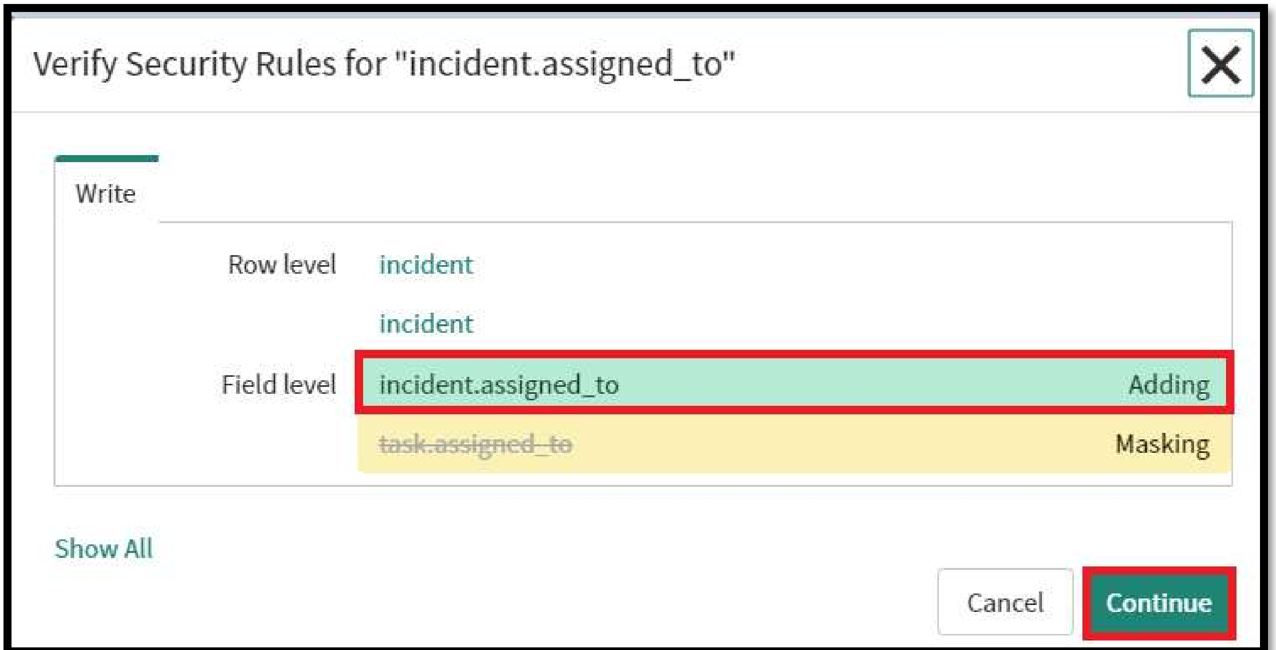
2. Navigate to **System Security > Access Control (ACL)**.



3. Click on **New**



4. **Type:** record
5. **Operation:** write
6. **Name:** [Incident] [assign_to]
7. **Role:** itil
8. **Condition:** [State] [is not] [Closed]
9. Click on **Submit**
10. Verify security rules when **Sumitted**



11. Click on **Continue**
12. Then check how it is working

Chapter – 21

Inbound Email Action

What is Inbound Email Action

Inbound email actions are enable to define the actions an instance takes when receiving email. inbound email actions define what actions Service Now takes when in response to incoming email.

1. Is the message a reply or forward?
2. Does the message match a record in the database?
3. Does the message match the conditions?

The system can take two types of actions:

1. **Record action:** We can set a value for a field in the target table.
2. **Email reply:** sending an email back to the source that triggered the action.

Types Inbound email action

The system classifies all incoming email into one of three types: forward, reply, or new.

New

The system classifies an email as **New** when it fails to match it to the forward and reply inbound action types.

Forward

The system classifies an email as a forward only when it meets all these criteria:

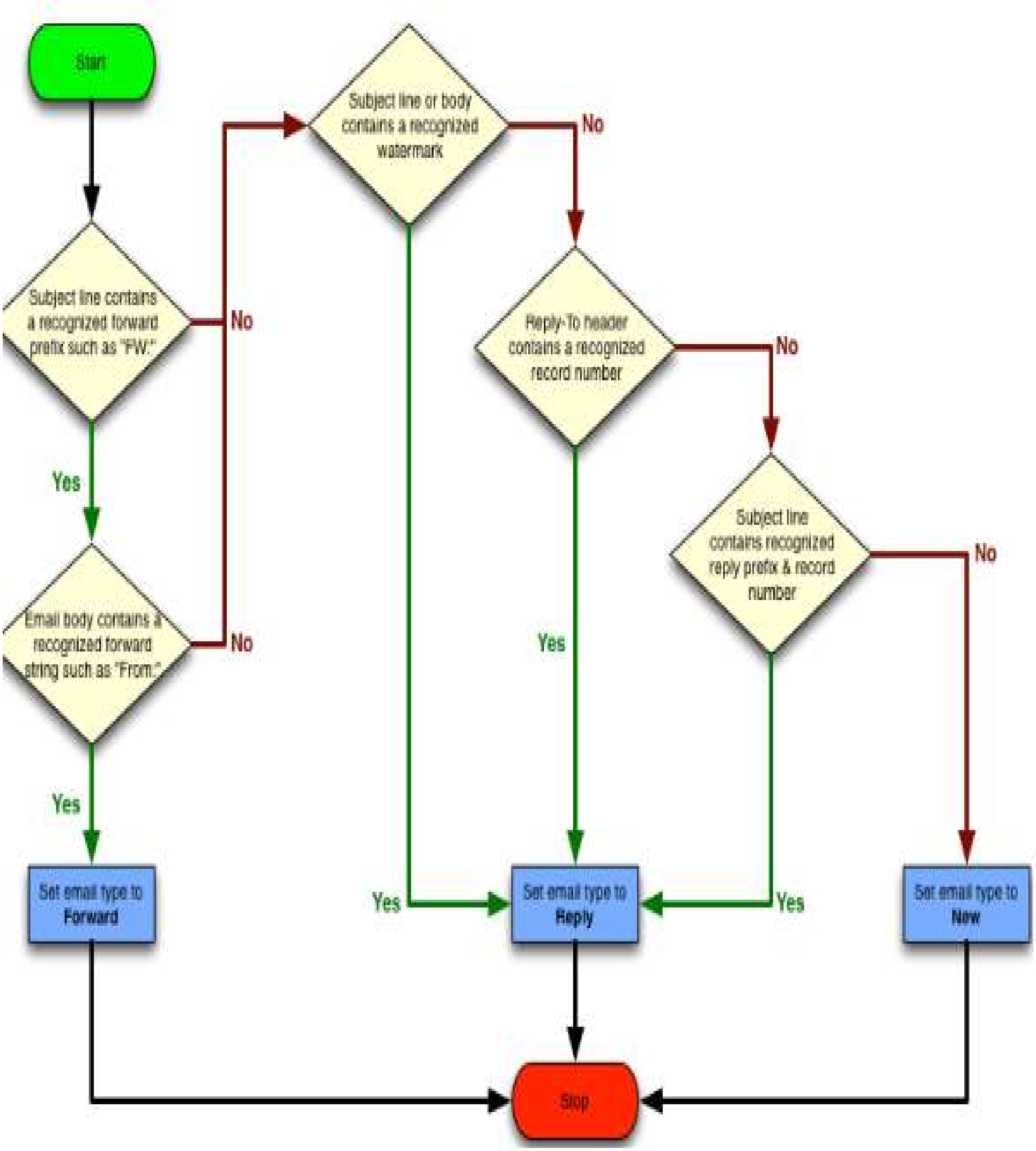
1. The subject line contains a recognized forward prefix such as **FW:**
2. The email body contains a recognized forward string such as **From:**
3. The system classifies any email that meets these criteria as a forward, even if the message contains a watermark or record number that otherwise classifies it as a reply.

Reply

The system classifies an email as a reply when it fails to match it to the forward inbound action type and it meets any one of these criteria:

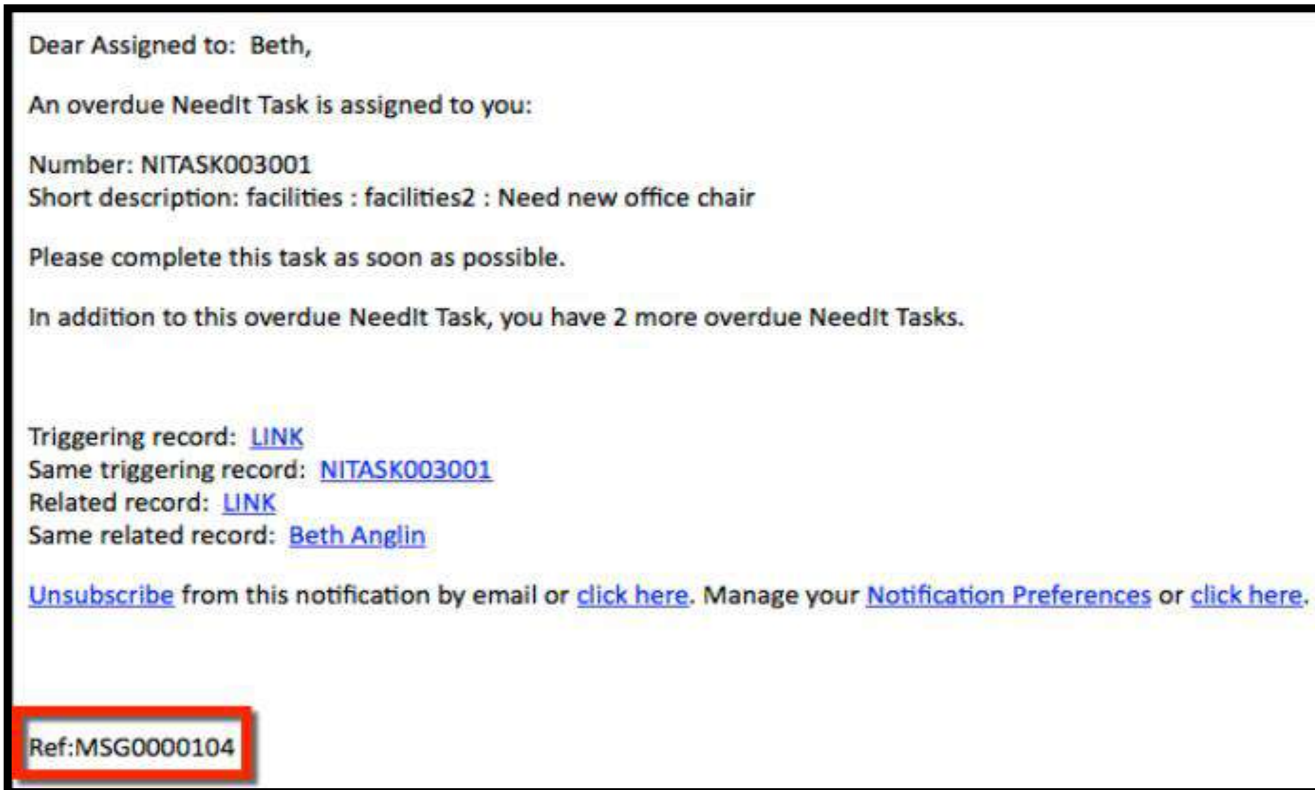
1. The subject line or email body contains a recognized watermark such as **Ref:MSG0000001.**
2. There is no watermark and the Reply-To header contains a recognized record number such as **INC0005552**
3. There is no watermark and the subject line contains a recognized reply prefix such as RE: and a recognized record number such as INC0005574

Determining the type of incoming email



How to send watermark or record number?

If an inbound message is recognized as a reply or forward ServiceNow attempts to match the incoming email with an existing record by looking for a unique watermark in the subject line or message body. By default, watermarks are inserted as the last line of outbound messages.



Dear Assigned to: Beth,

An overdue NeedIt Task is assigned to you:

Number: NITASK003001
Short description: facilities : facilities2 : Need new office chair

Please complete this task as soon as possible.

In addition to this overdue NeedIt Task, you have 2 more overdue NeedIt Tasks.

Triggering record: LINK
Same triggering record: NITASK003001
Related record: LINK
Same related record: Beth Anglin

Unsubscribe from this notification by email or click here. Manage your Notification Preferences or click here.

Ref:MSG0000104

If no watermark is found, ServiceNow searches the subject and body for a record number.

Conditions

The final step in determining which Inbound Action executes is to check the conditions for a match. Use the Condition builder to create the condition or write a condition script. If there is no value in a condition field, the field returns true.

Inbound Email Actions execute only when:

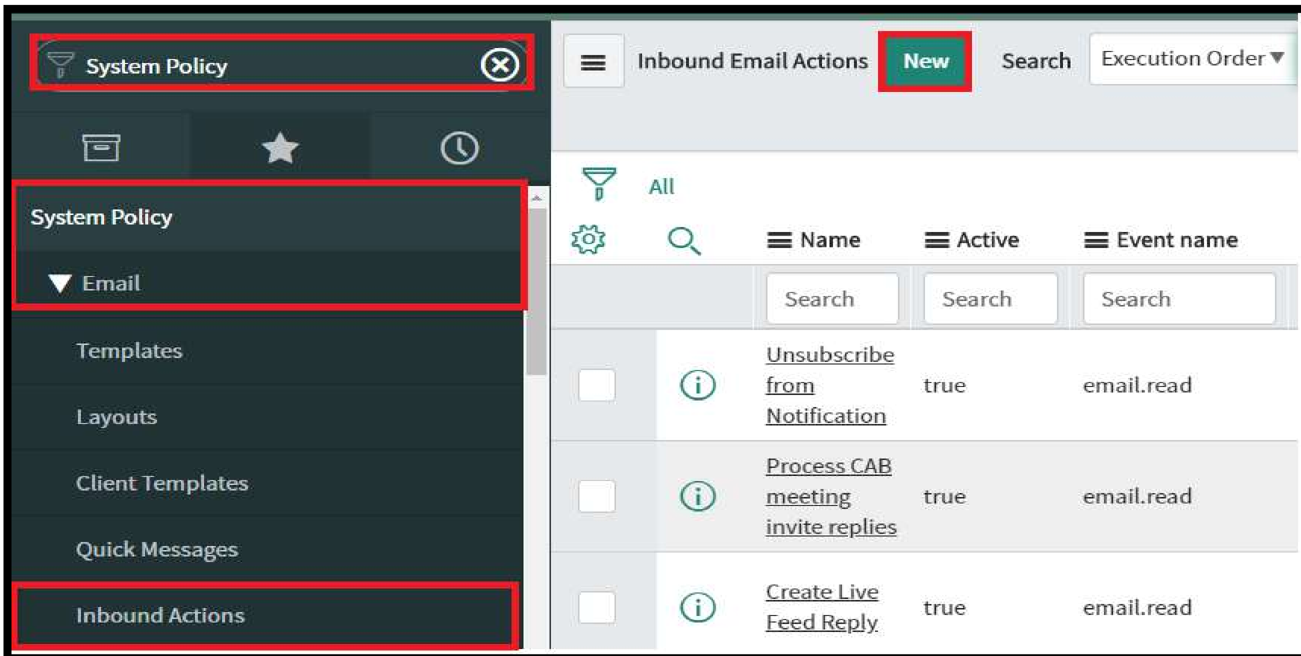
1. The incoming email type matches the inbound action type
2. The watermark or record number matches a record on the target table
3. The inbound action Conditions evaluate to true

Exercise: 1 Creating Inbound Email Actions

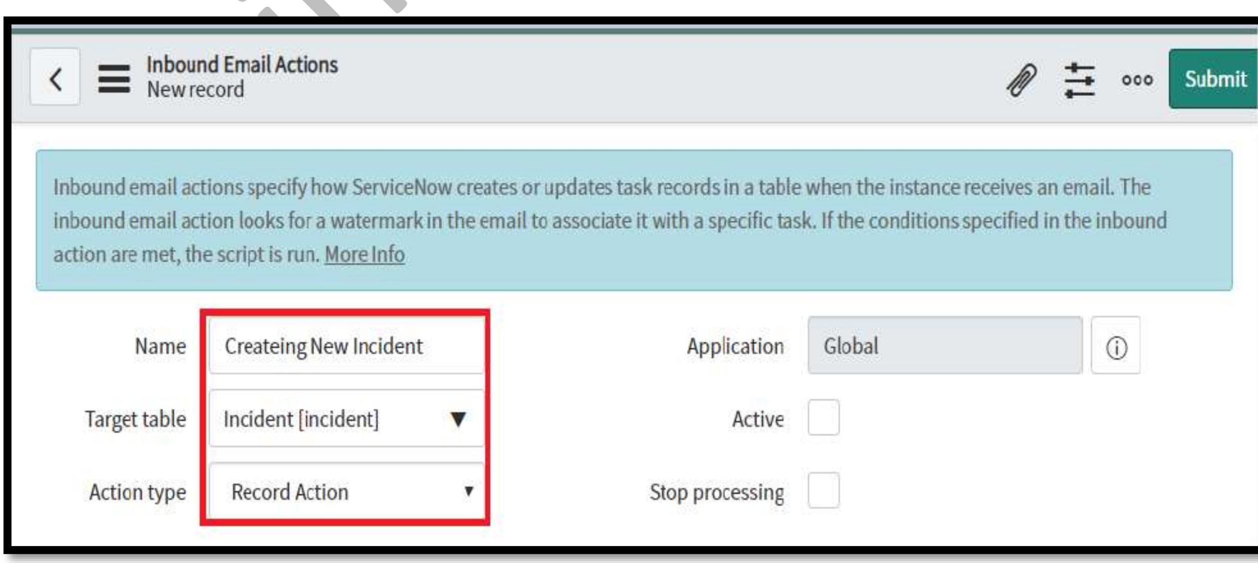
We are going to create an inbound email action to create an incident automatically in incident table when a customer sends an email

Procedure

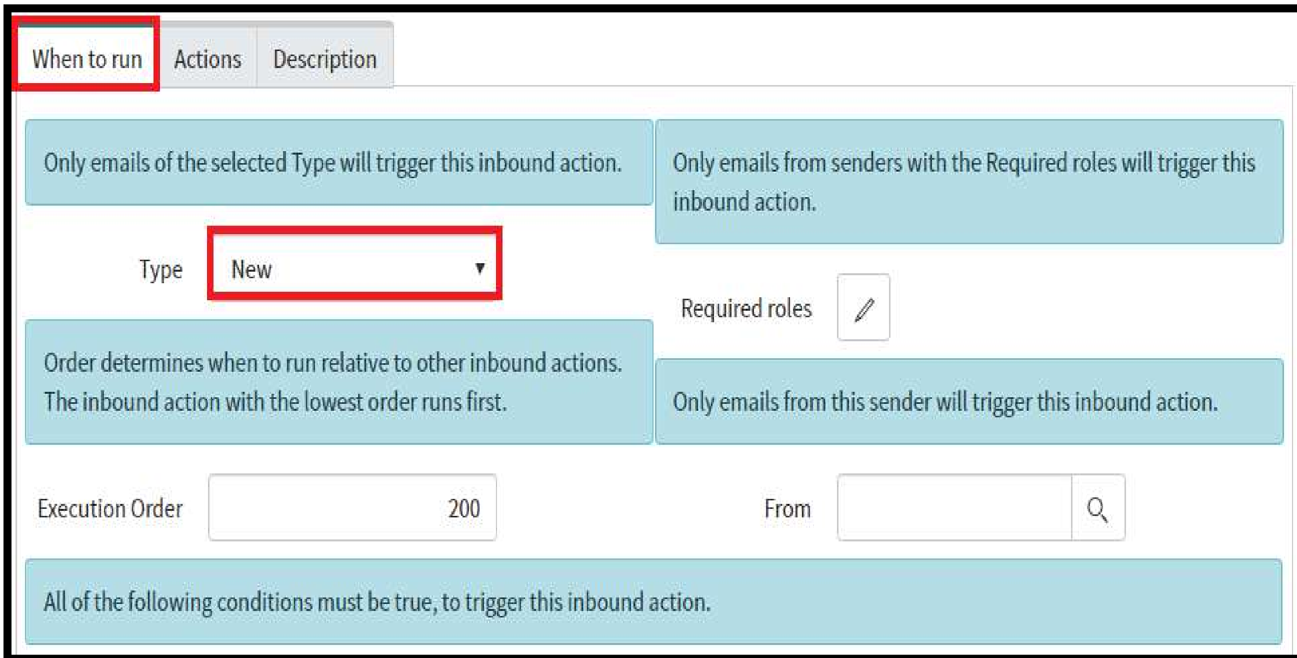
1. **System Policy > Email > Inbound Action**
2. **Click on New**



3. Fill **Inbound Email Action** form



4. **Name:** Creating New Incident
5. **Target Table:** Incident
6. **Action Type:** Record Acton



7. Open **When to run** Tab
8. **Execution Order:** 200

When to run | **Actions** | Description

```
//  Note: current.opened_by is already set to the first UserID that matches the From: email address

current.caller_id = gs.getUserID();
current.comments = "received from: " + email.origemail + "\n\n" + email.body_text;
current.short_description = email.subject;

current.category = "inquiry";
current.incident_state = IncidentState.NEW;
current.notify = 2;
current.contact_type = "email";

if (email.body.assign != undefined)
   current.assigned_to = email.body.assign;

if (email.importance != undefined) {
   if (email.importance.toLowerCase() == "high") {
        current.impact = 1;
        current.urgency = 1;
   }
}

current.insert();
```

9. Write code like Above screen shot
10. Click on **Submit**

Chapter - 22

Reports

What is Report

Service Now platform can be providing reporting capabilities that we can create and sharing reports that defines the current state of instance data, such as how many open incidents of each priority there are and how many in active. Reporting functionality is available by default for all tables.

Service Now generating and displaying reports in visualizations manner of our data that you can share with users on dashboards and service portals, we can be able to export to PDF, and send the reports through email. Now we have to understand how to **create, run, edit, view, and share** reports.

Types of Reports

We can create different types of reports and share them into acros all people in our organization based on our client requirement

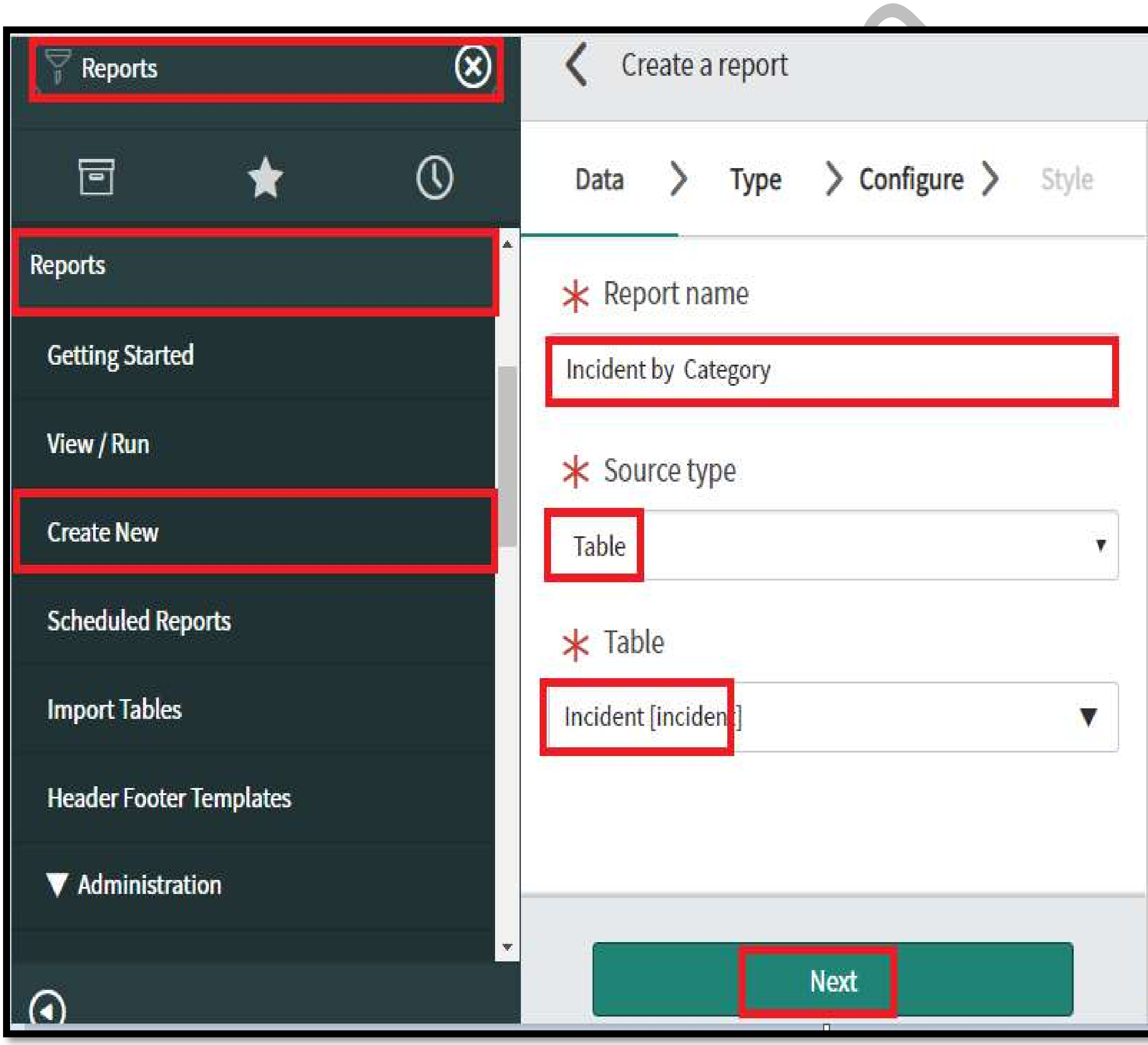
1. List
2. Bar Charts
3. Pie Charts
4. Pareto
5. Histogram
6. Pie
7. Speedo Meter
8. Dial
9. Pivot
10. Box
11. Pyramid
12. Trend
13. Donuts
14. Area
15. Heat map
16. Map
17. Booble
18. More…

Exercise: 1 Creating a new Report

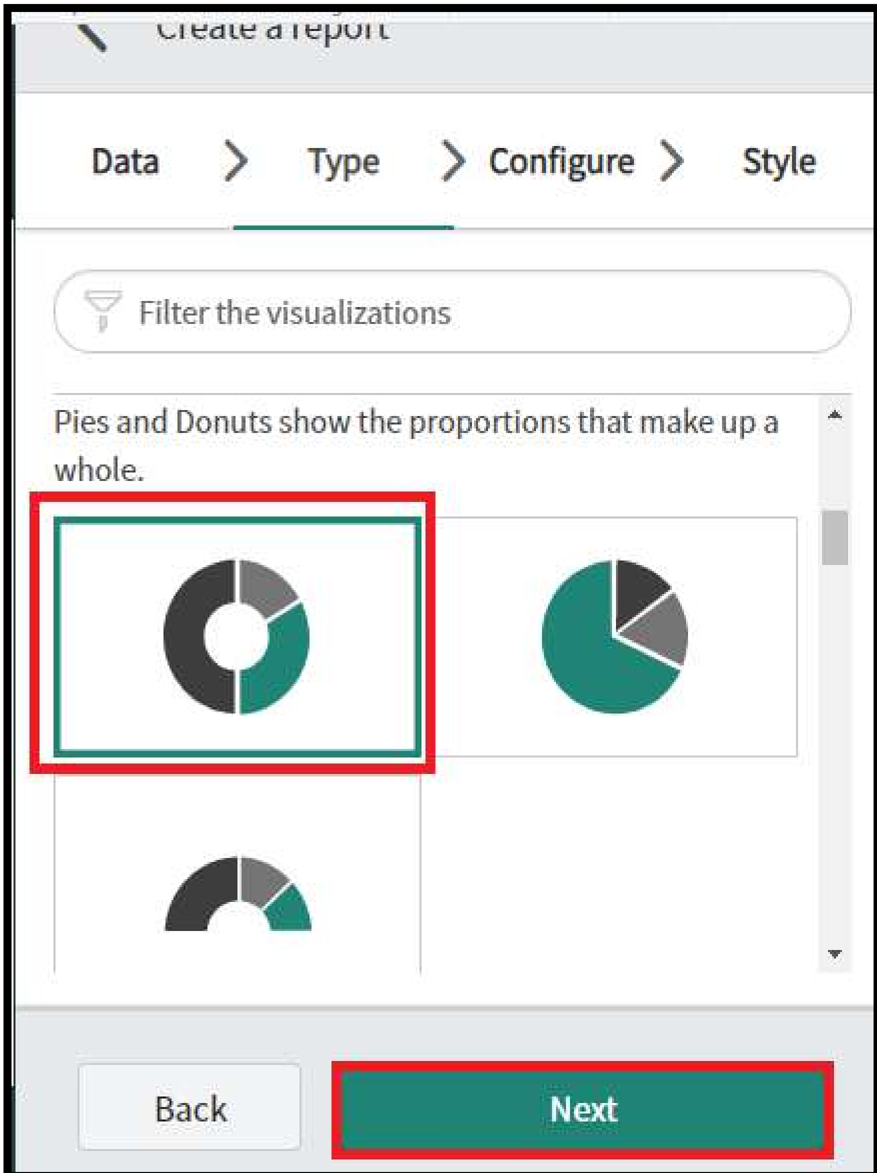
We are going to creating new report and share to respective network group

Procedure

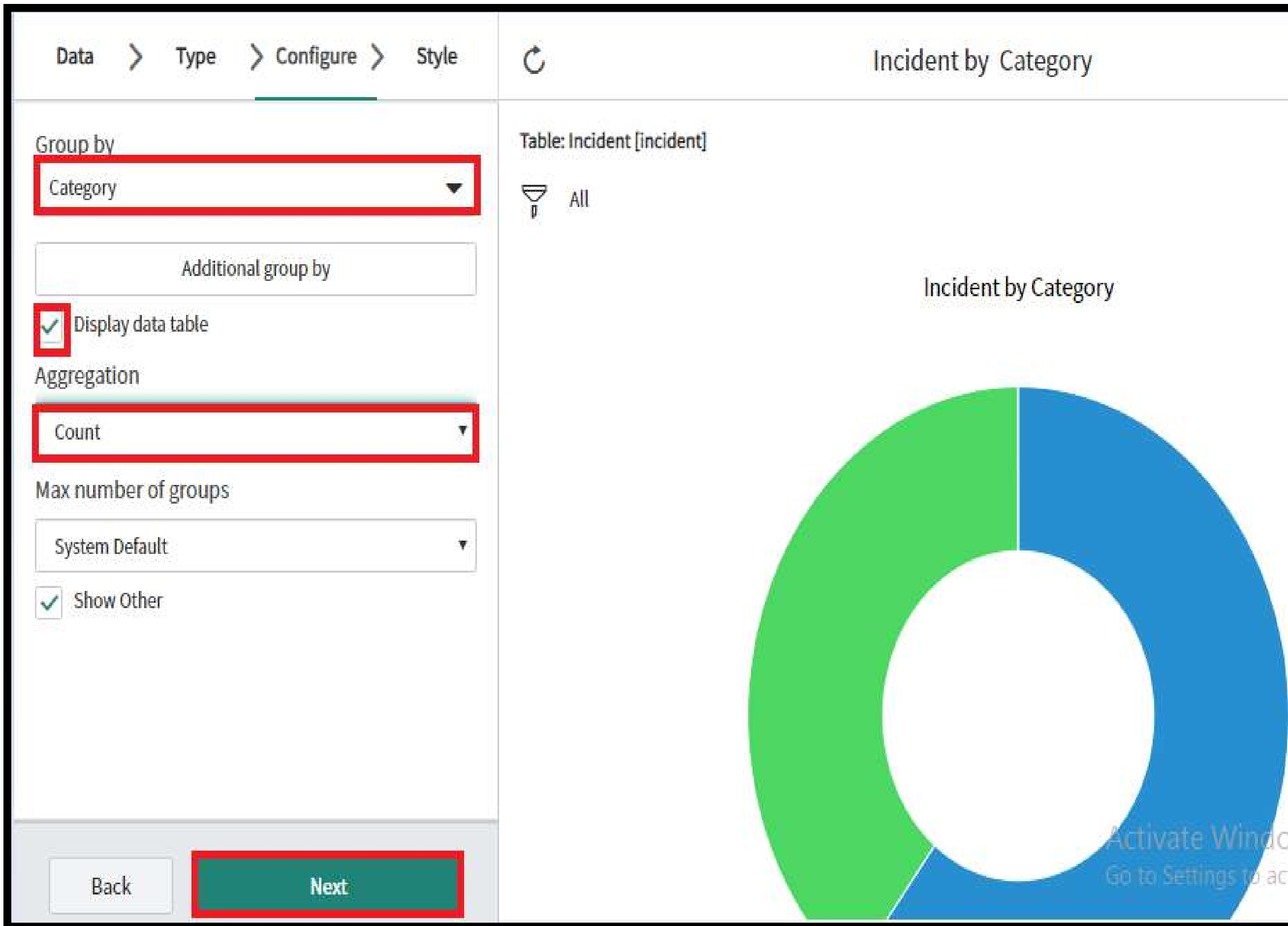
1. Navigate to **Reports > Crete New**
2. **Report Name:** Incidents by category
3. **Source Type:** Table
4. **Table:** Incident



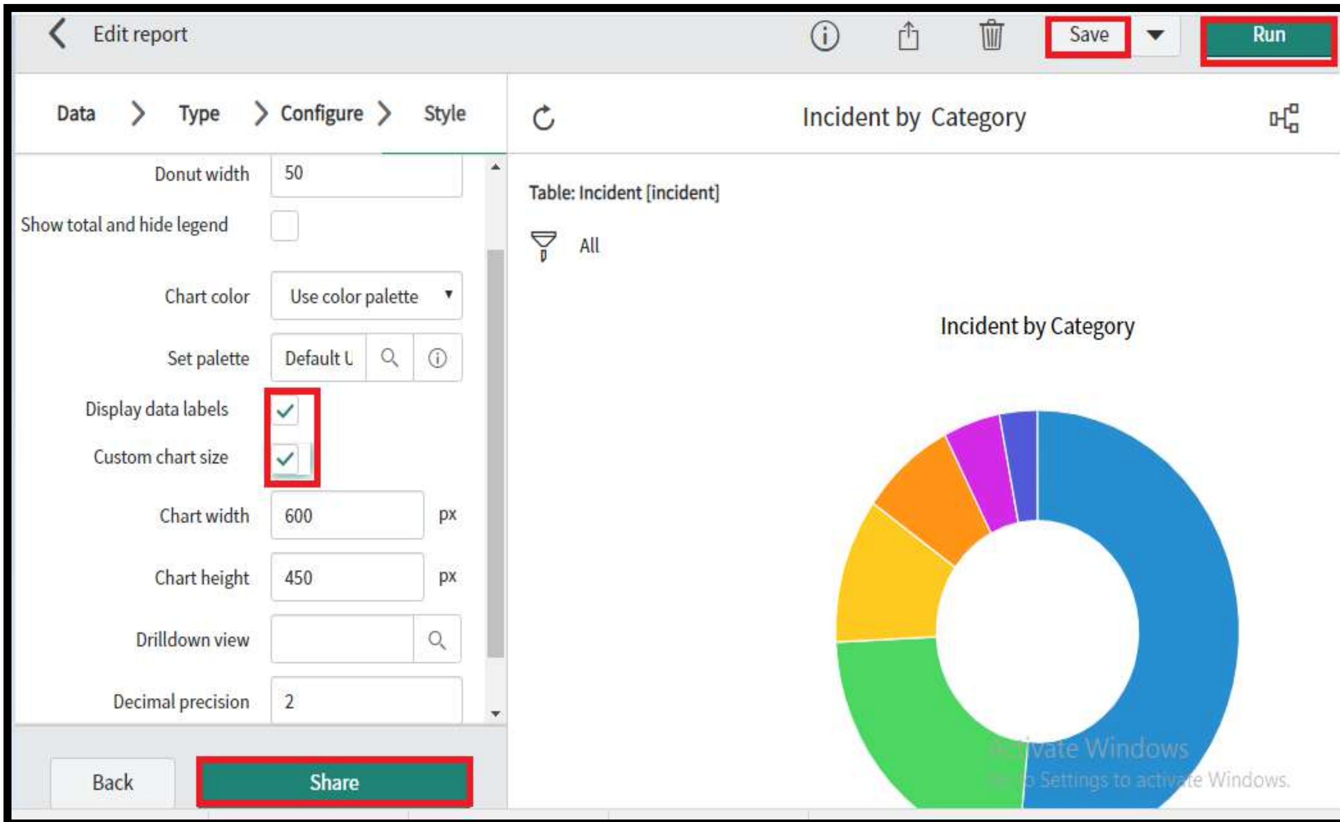
5. Click on **Next Button**



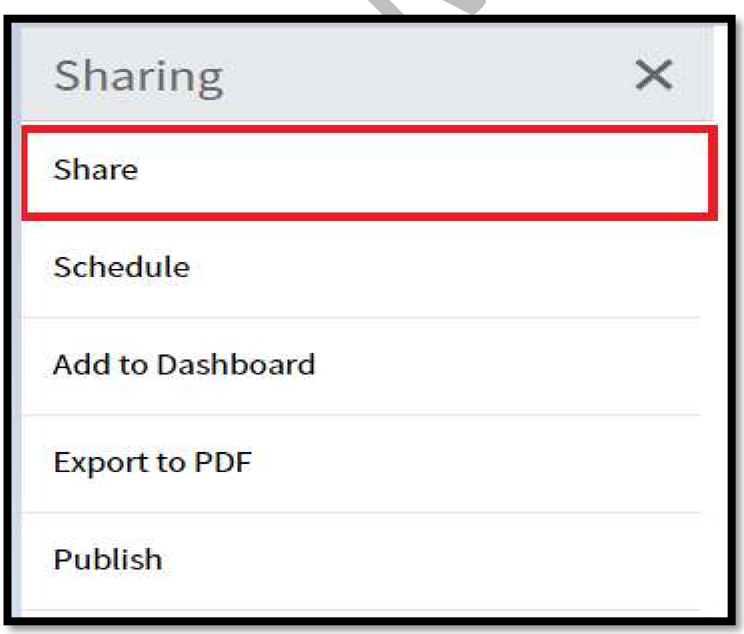
6. Select **Pie chart**
7. Click on **Next** Button



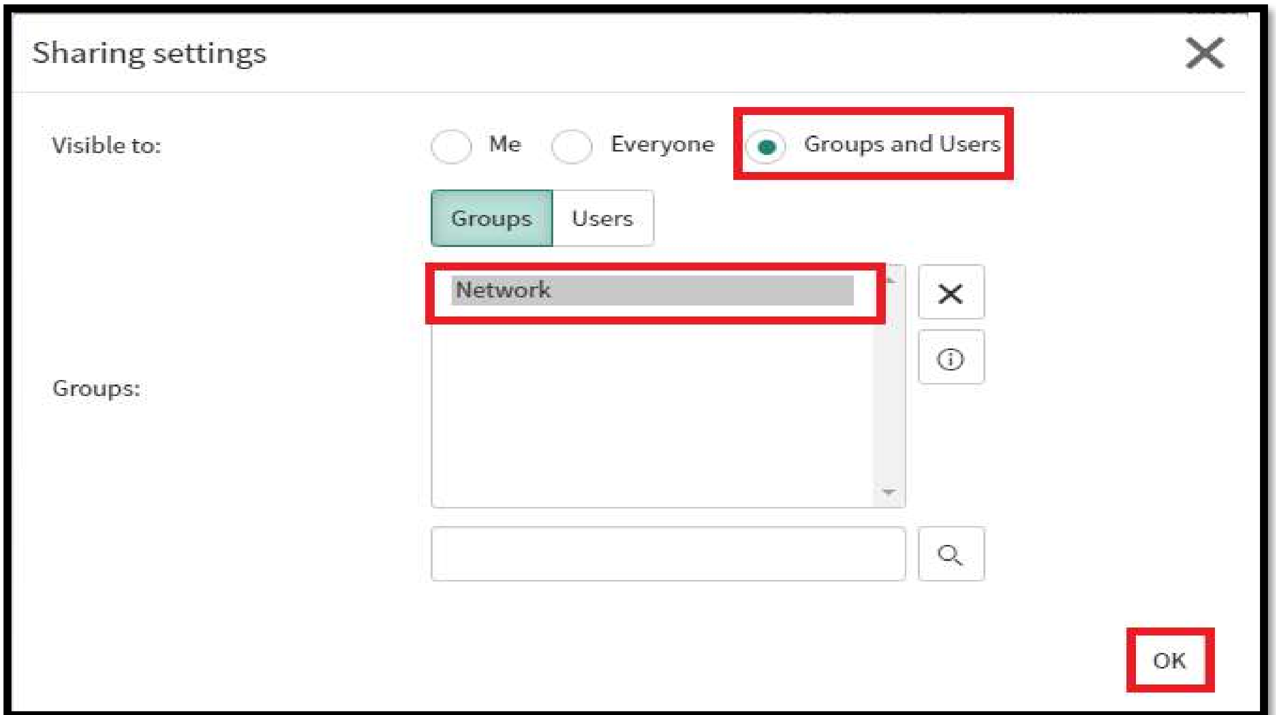
8. **Group by:** category
9. **Display Data table:** checked = true
10. **Aggregation:** Count
11. **Max num of groups:** system default
12. Click on **Next**



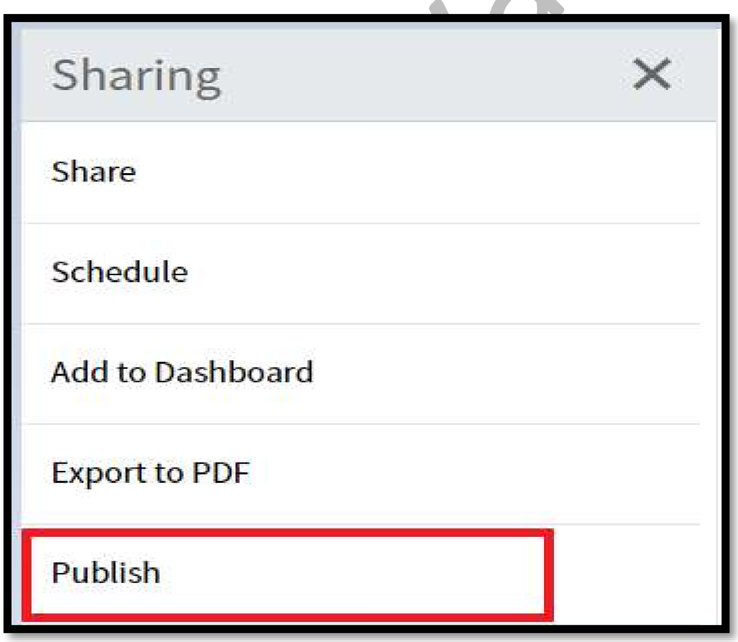
13. Display Data labels: checked true
14. Custom chart table: checked true
15. Click on **Save**
16. Click on Click on **Share Button**



17. Click on **Share**



18. Visible to: Group and Users
19. Groups: Network
20. Click on **OK**



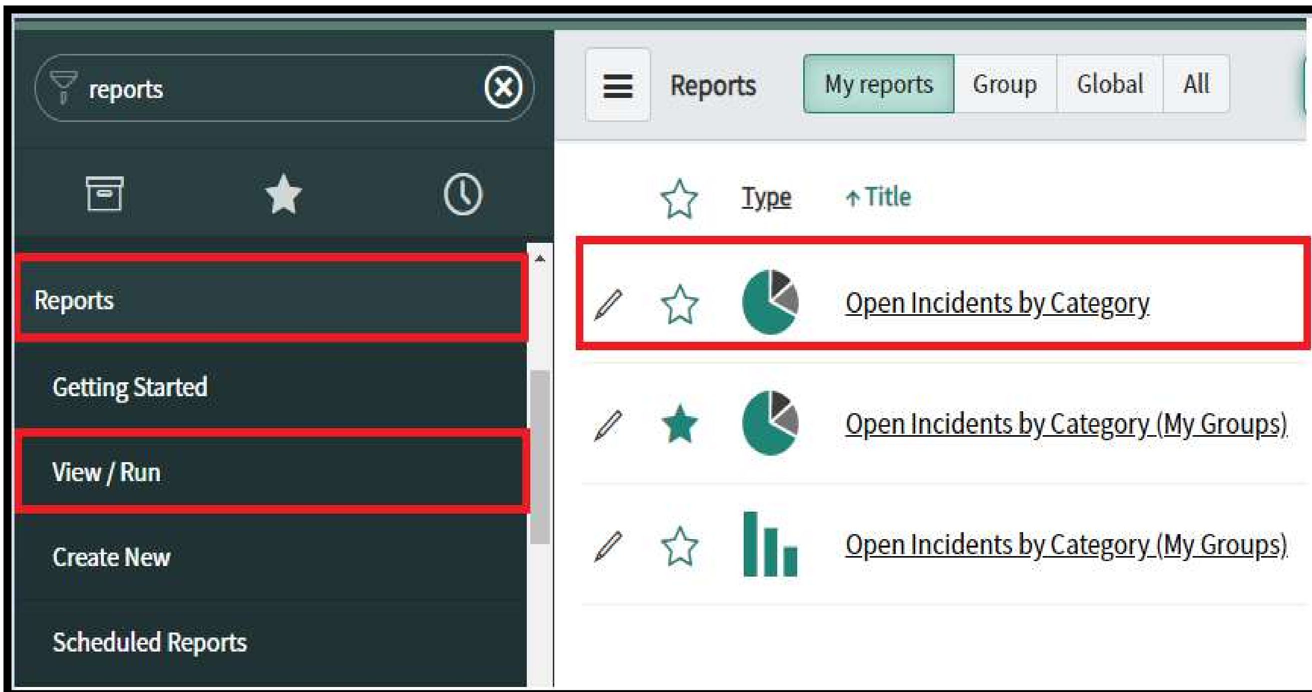
21. Click on **Publish**

Schedule and sharing report

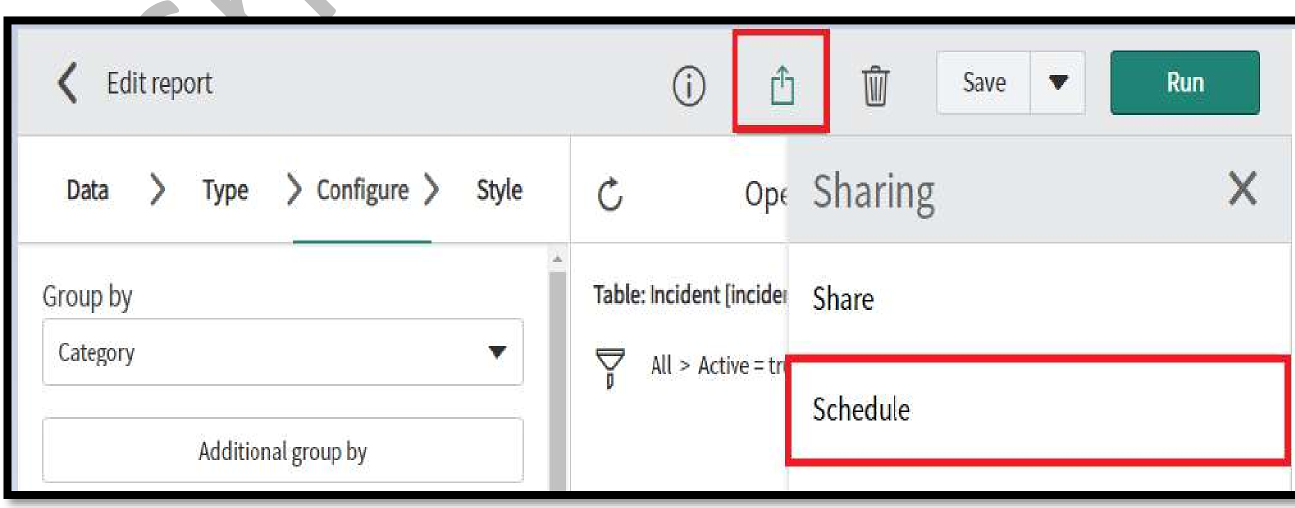
We are going to schedule and automate the distribution of reports, schedule reports will help us to send reports in visualization manner regularly in our emails to different groups and users

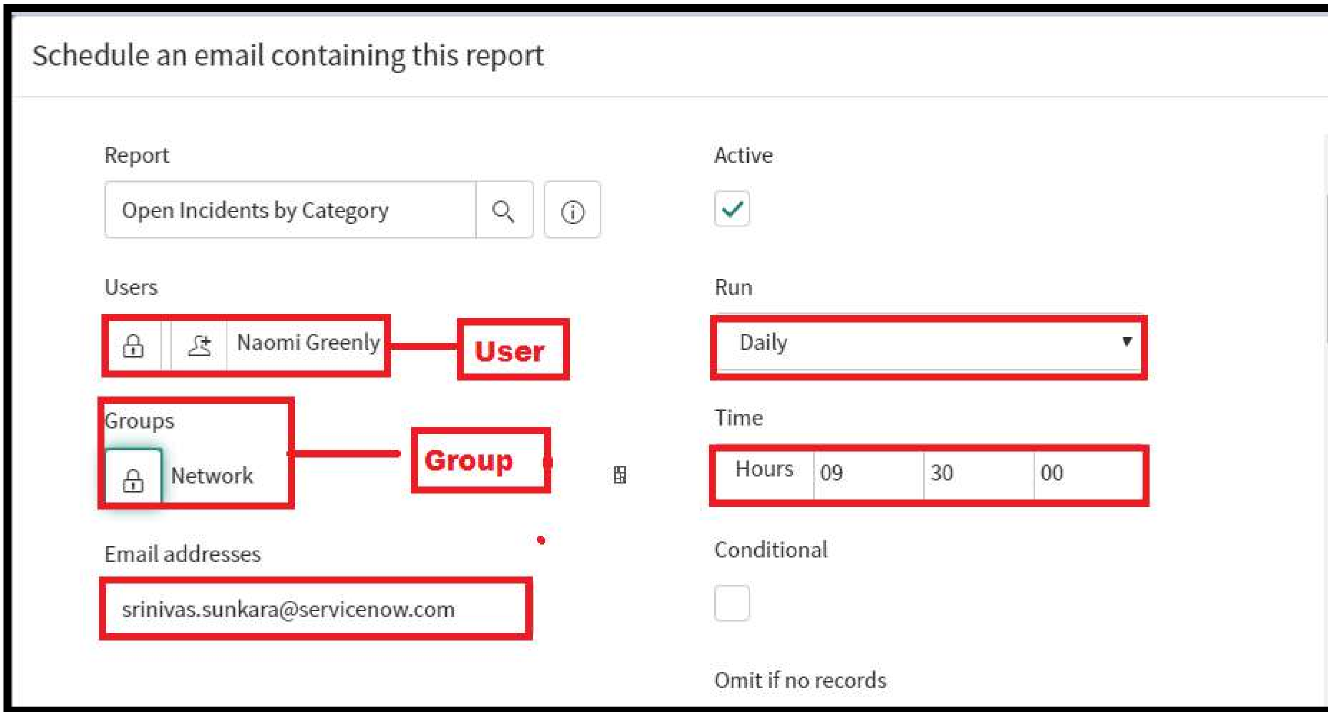
Procedure

1. Navigate to **Reports >** Click on **View/Run.**
2. Select our existing report to schedule for distribution.

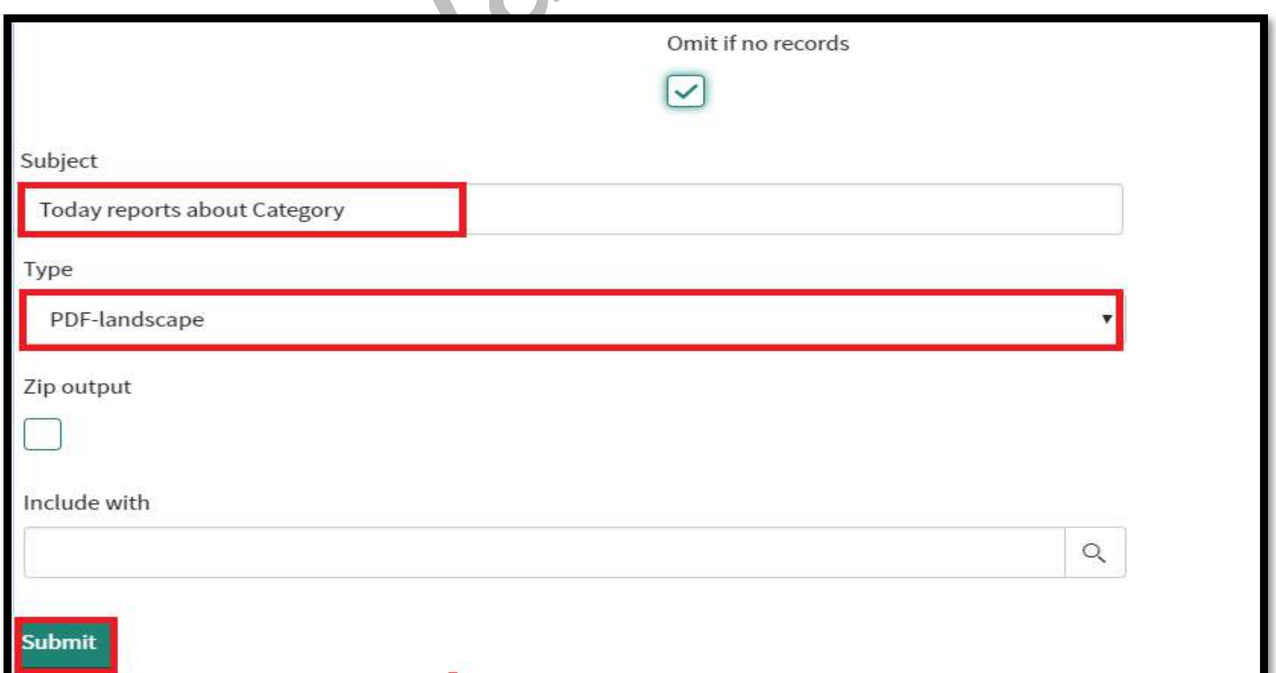


3. Click the arrow next to the Save button to open the Report options menu and

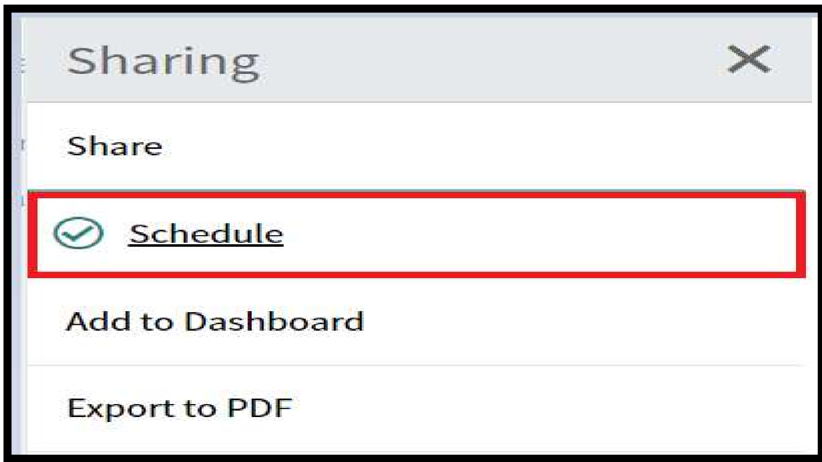




4. **User:** Select any user to send report
5. **Groups:** Select any group to send report
6. **Select Addresses:** provide any other user mail id to send report
7. **Run:** daily
8. **Time:** 9:30 AM



9. **Subject:** Today reports about category
10. **Type:** PDF-land scape
11. Click on **Submit**



12. **Check** scheduled report

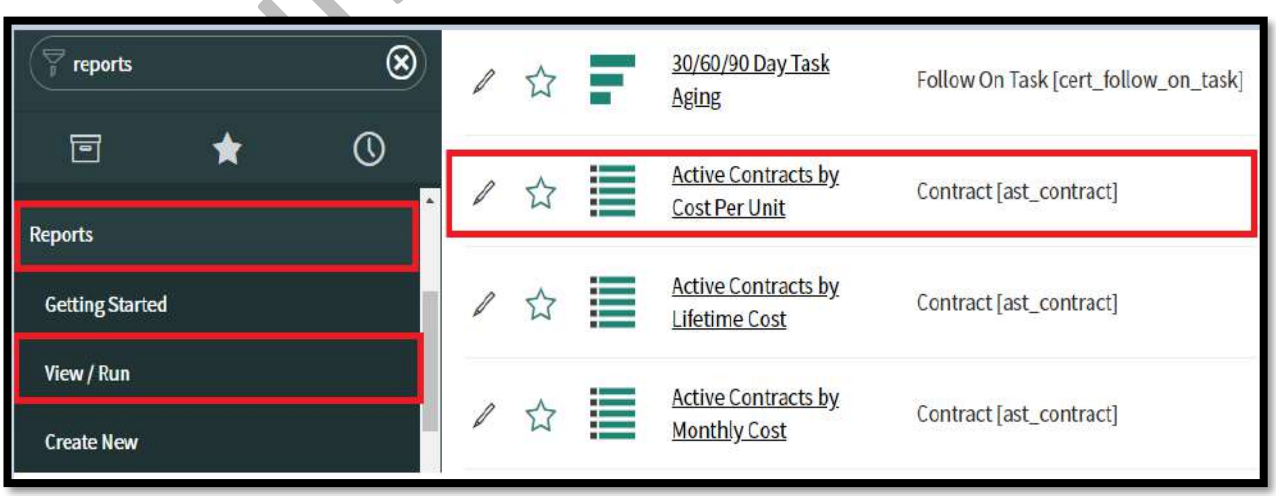
Delete a report

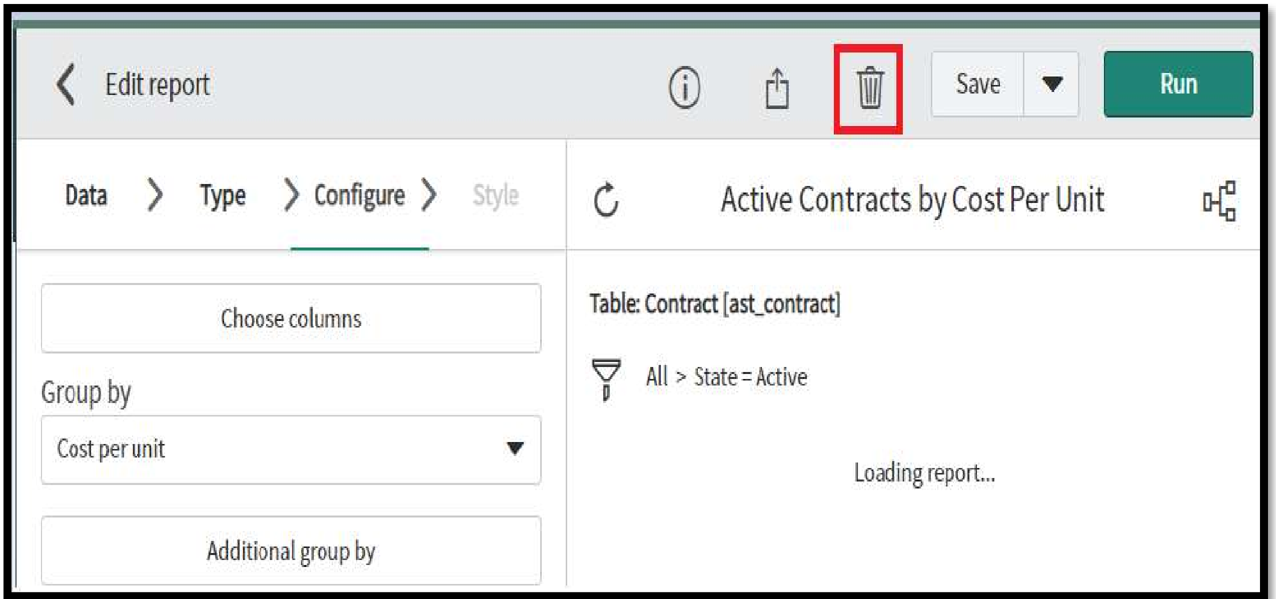
We are going to delete reports those are no longer used in our instance.

Note: Only report creator or snow administrator of the report to delete it

Procedure

1. Navigate to **Reports > View / Run.**
2. Select the report to delete.





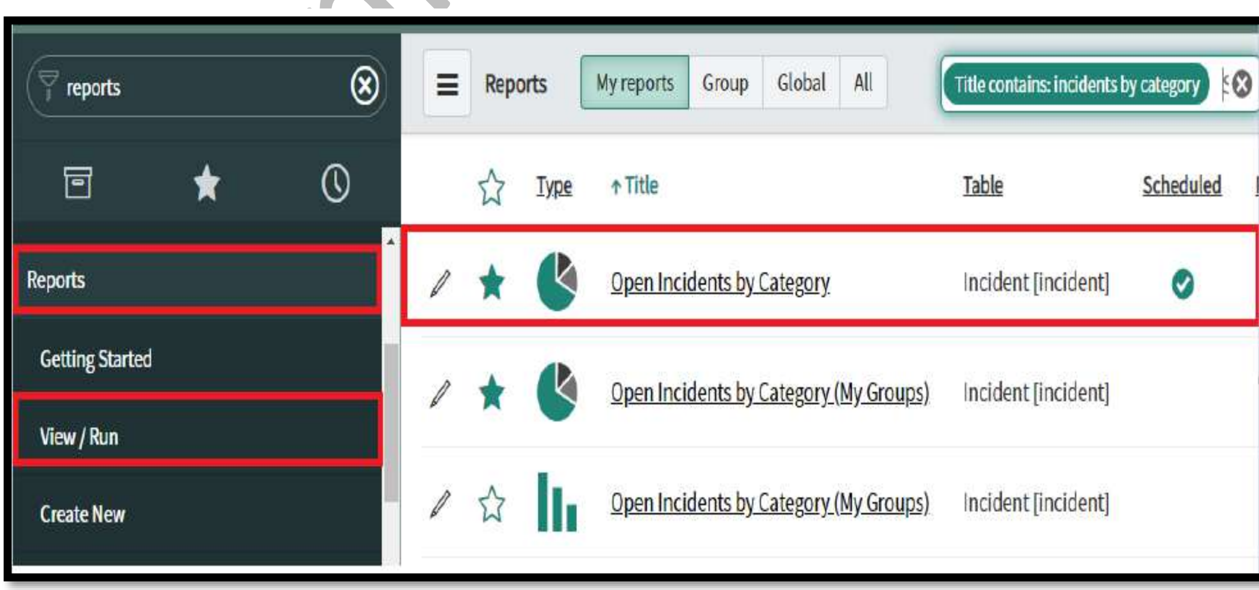
3. When the report opens, click the **Delete icon** and confirm that you want to delete the report.

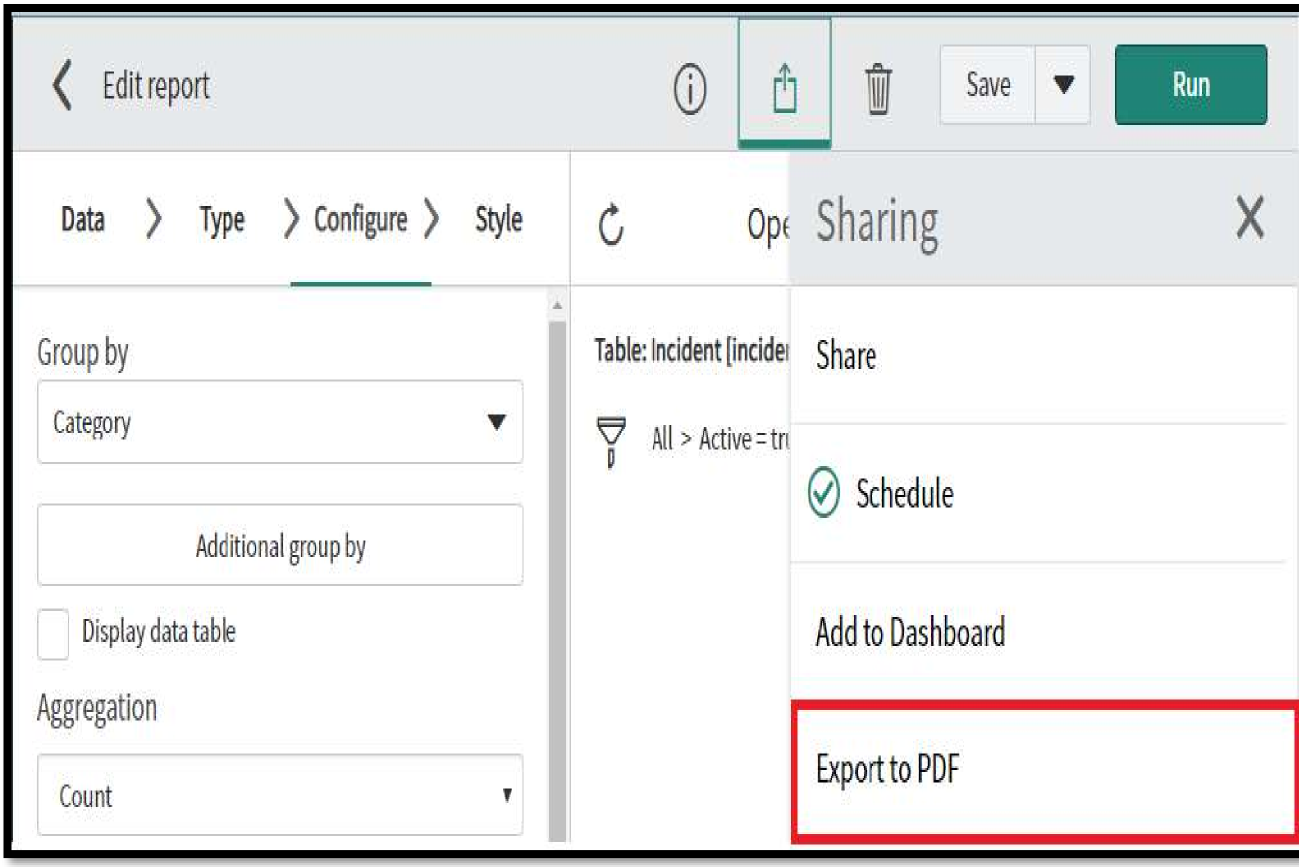
Export report to PDF

We are going to exporting our reports to PDF file and share through mail

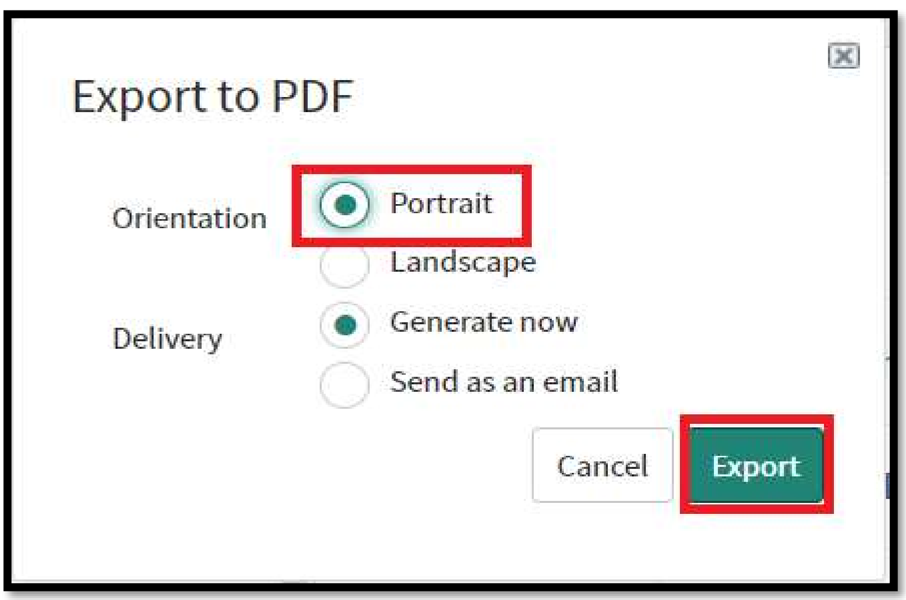
Procedure

1. Navigate to **Reports > View / Run.**
2. Select the report to **Export PDF.**

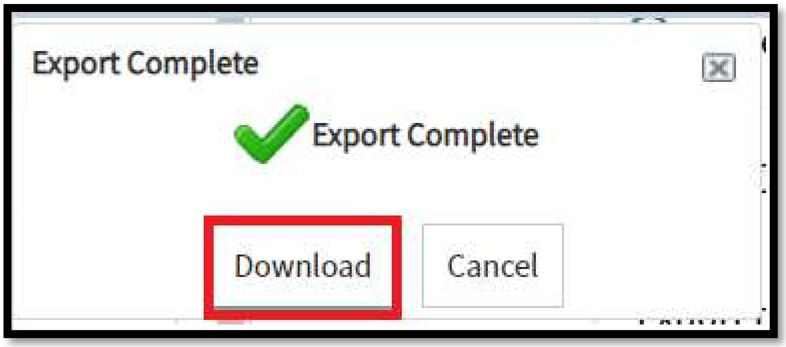




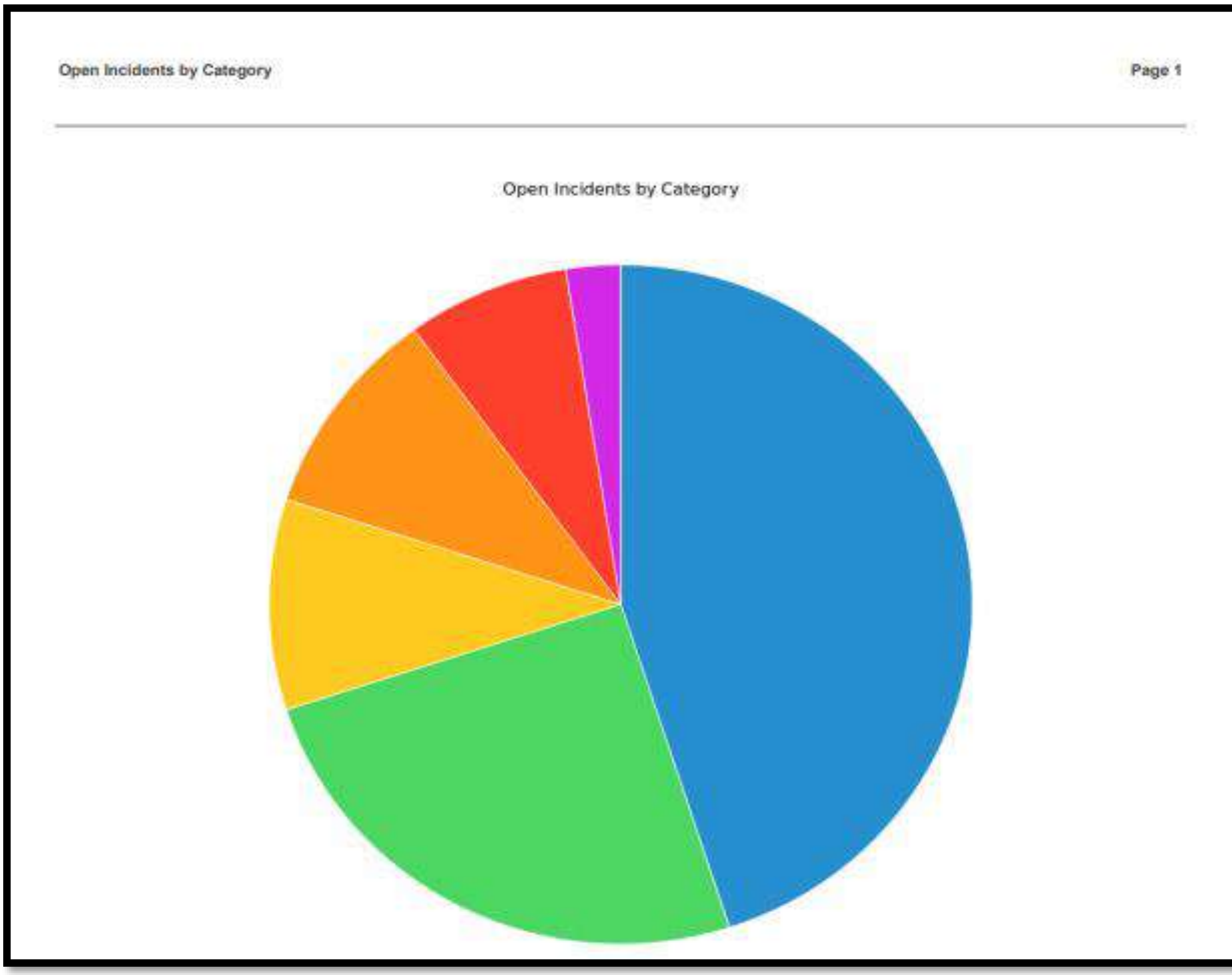
3. Click on **Export to PDF**



4. **Orientation:** Portrait
5. Click on **Export Button**



6. Click on **Download** Button



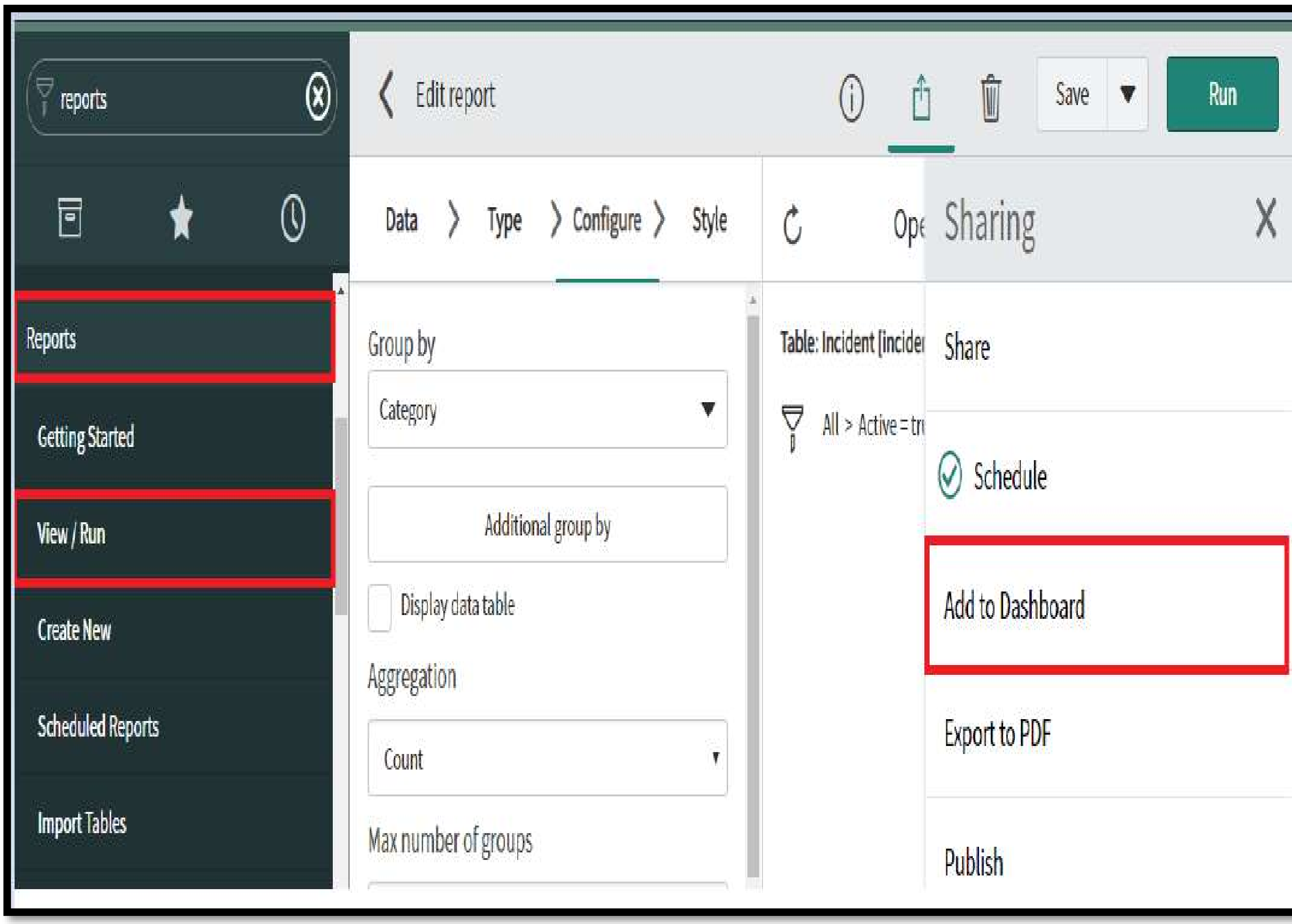
7. Check the result in **PDF File**

Report add to Dashboard

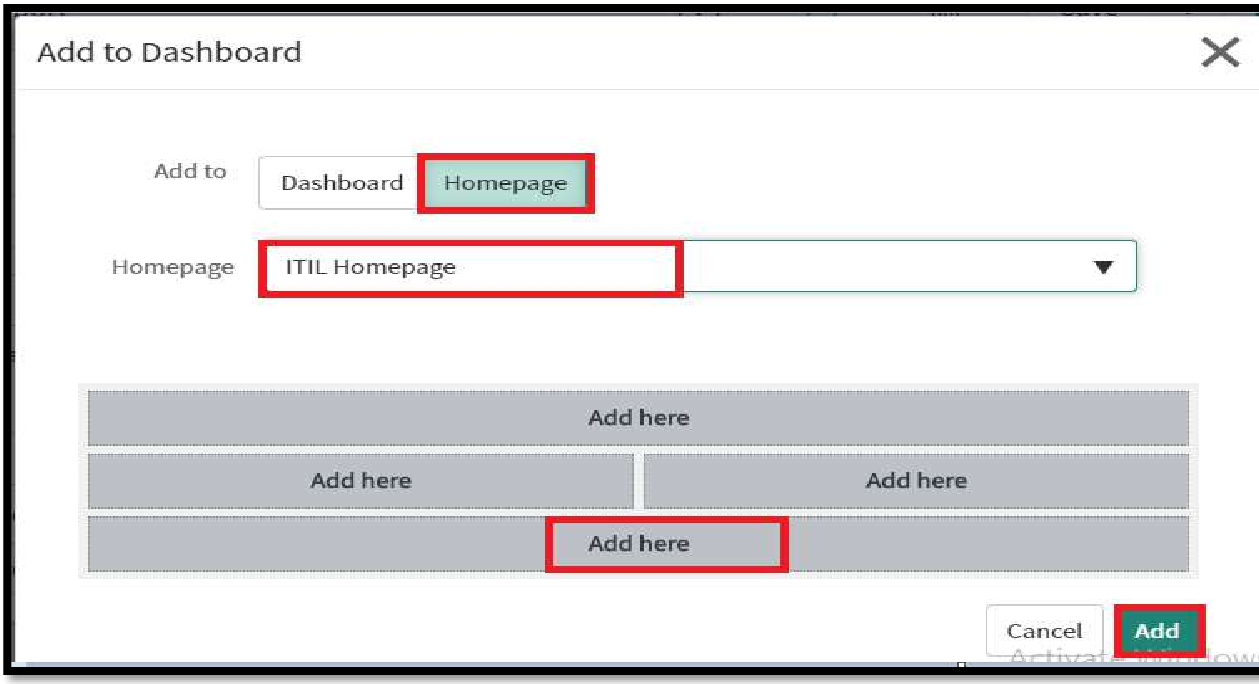
We are going to add a report to our dashboard for self-service, itil, admin interfaces

Procedure

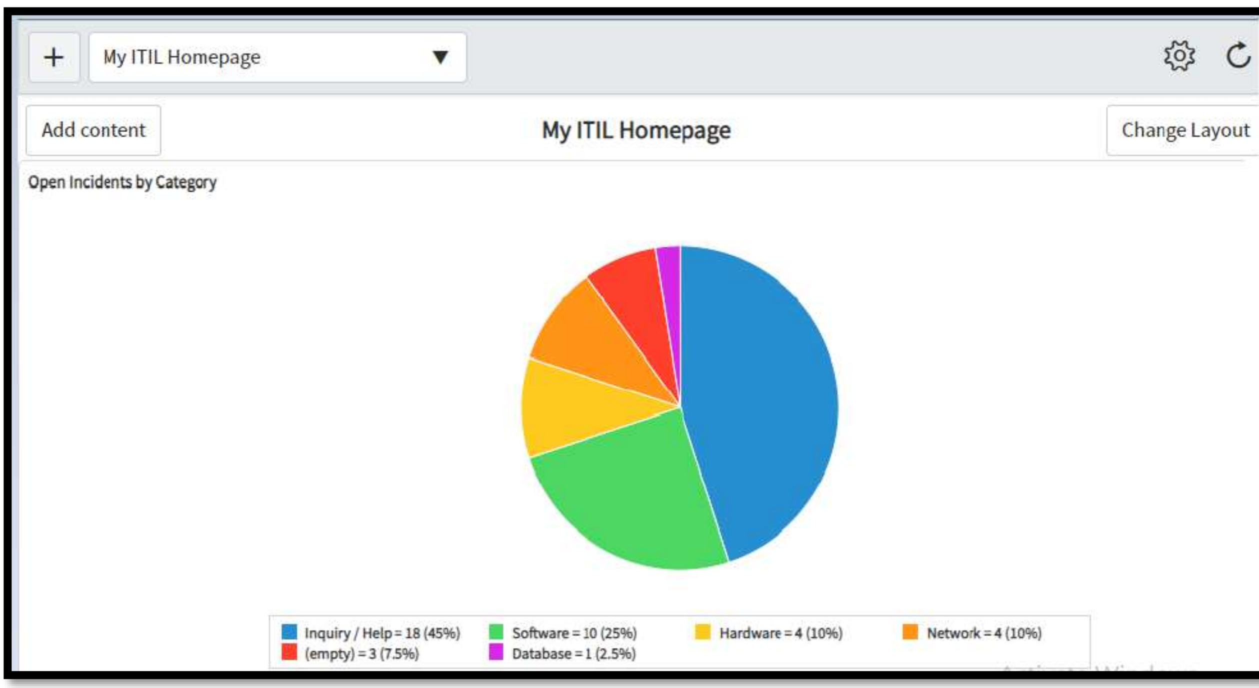
1. Navigate to **Reports > View/Run**
2. Select report to add your **Dashboard**
3. Click on Add to **Dashboard**



4. Select **Home Page** tab
5. **Home page:** ITIL Home Page
6. Click on **Add Here**



7. Click on **Add Button**



Second Edition release soon…