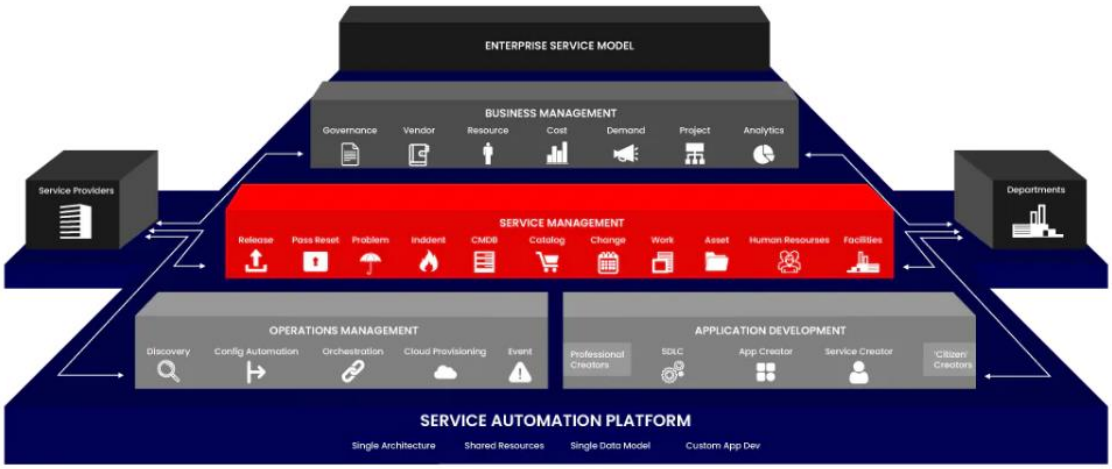
Understanding document of ServiceNow Administration Fundamentals #Week - 2

The **ServiceNow Platform** is a cloud-based Application Platform-as-a-Service (APaaS), providing companies with a streamlined solution to avoid managing their own hardware. ServiceNow platform provides enterprise IT operations management, automating various business processes through workflows and applications. It is used across industries to manage IT service delivery, human resources, customer service, and security operations.

Platform Architecture

- **Multi-instance, single-tenant architecture**: Each instance has an isolated database containing all data, apps, and customizations.
- **Flexible web-based user interface**: Built on a dynamic database schema, allowing easy configuration.
- **Single system of record**: Consolidates business processes into a unified structure.
- **Integration**: Easily connects with other enterprise systems, supporting plug-and-play applications.
- **Custom application development**: Enables users to build and host custom applications on the platform.
- **Data redundancy**: Every ServiceNow data center is paired with another to ensure backup and failover across all layers of infrastructure.
- **Backups & Security**: ServiceNow conducts four weekly full data backups and daily differential backups, secured through third-party organizations.



Authentication in ServiceNow

Authentication is the process that ensures only authorized users can access the ServiceNow platform. ServiceNow provides several methods for authentication, ranging from basic local database authentication to more secure methods like Single Sign-On (SSO) and Multi-factor Authentication (MFA). Local authentication involves storing user credentials in the platform's database, while SSO enables users to log in using credentials from external systems such as Microsoft Azure, Okta, or other identity providers. This flexibility ensures that organizations can adopt an authentication method that fits their security requirements.

Methods of Authentication:

1. Local database authentication
2. External Single Sign-On (SSO)
3. Multi-factor authentication
4. Digest token authentication
5. OAuth 2.0

- **Local Database Authentication**: ServiceNow also supports local authentication, where users authenticate directly with the platform using credentials stored within ServiceNow. Users log in using credentials stored in the ServiceNow instance itself.
- **External Single Sign-On (SSO):** This allows users to authenticate once and gain access to all the applications they are authorized for without logging in again, providing a seamless user experience. SSO in ServiceNow can be integrated with third-party identity providers like SAML 2.0, OAuth, and OpenID Connect.
- **Multi-Factor Authentication (MFA):** MFA adds an extra layer of security by requiring users to provide multiple forms of verification, such as a password and a one-time token sent to their mobile device. This ensures that even if a password is compromised, unauthorized access is less likely.
- **LDAP Authentication:** Integration with Lightweight Directory Access Protocol (LDAP) enables organizations to centralize authentication using their existing LDAP infrastructure. This allows ServiceNow users to authenticate using the same credentials they use for other enterprise systems.

- **OAuth 2.0**: Allows authentication using third-party OAuth providers for secure access.

Authentication mechanisms in ServiceNow are vital for maintaining secure access to sensitive information, ensuring that users are who they claim to be before granting them access.

Types of Instances

1. **Production Instance -** The live, operational instance.
2. **Non-Production Instances** - Development, testing, quality assurance
   ServiceNow provides a **Personal Developer Instance (PDI)** for learning and experimentation.
   Load-balanced instances are hosted in ServiceNow Data Centers, but can be implemented on-site for some customers.
3. **Personal Developer Instance (PDI)**: A free, individual instance for learning and experimentation.

Instances are hosted in ServiceNow Data Centers globally, with unique URLs formatted as https://<instance name>.service-now.com.

User, Group, Role in ServiceNow

ServiceNow organizes access to its resources based on users, groups, and roles, which allows for granular control over permissions:

- **User**: A user represents an individual who interacts with the ServiceNow instance. Users are assigned specific roles that grant them access to different parts of the platform. Each user has a profile containing attributes like their name, department, and contact information.

  An individual interacting with ServiceNow functionalities. Stored in the sys_user table.

- **Group**: A collection of users sharing common characteristics for role assignment, access controls, etc. For example, an IT support group might include all IT technicians who need access to incident management applications. Users can belong to multiple groups, and groups can be assigned specific roles.

  Stored in the sys_user_group table.

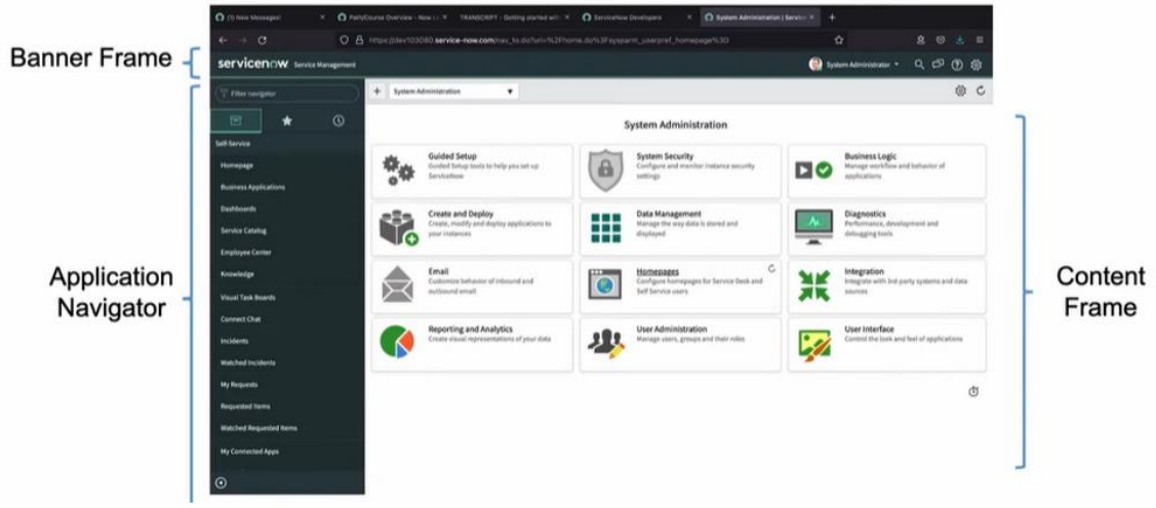
- **Role**: Roles define the permissions for what a user or group can access and perform within ServiceNow. Roles are used to control access to applications, modules, and features like creating records, editing data, or running reports. By combining users, groups, and roles, organizations can effectively manage access controls.

  Stored in the sys_user_role table.

This user management framework allows for efficient delegation of tasks, ensuring that only authorized personnel can access or modify data, which is especially important in regulated industries like healthcare and finance.

ServiceNow User Interface Overview

The ServiceNow User Interface (UI) is designed to be intuitive, enabling users to navigate the platform effortlessly. The interface includes key components such as the Application Navigator, which allows users to browse applications, and the content frame, where data and forms are displayed. The UI is customizable, so users can modify their dashboards, menus, and themes according to personal or organizational preferences. System administrators also have the capability to control UI branding and appearance for consistency across all users in the organization.



Key UI Components:

1. **Application Navigator**: The left-hand menu that gives access to various applications and modules.
2. **Content Frame**: The central display area where forms, lists, and dashboards appear.
3. **User Menu**: Located in the upper right, this menu provides options to manage settings, switch users, or log out.
4. **System Settings**: Enables users to personalize their display preferences, such as language and theme.
5. **Banner Frame**: Displays important notifications and branding elements, customizable to reflect an organization's identity.

The interface has three main parts:

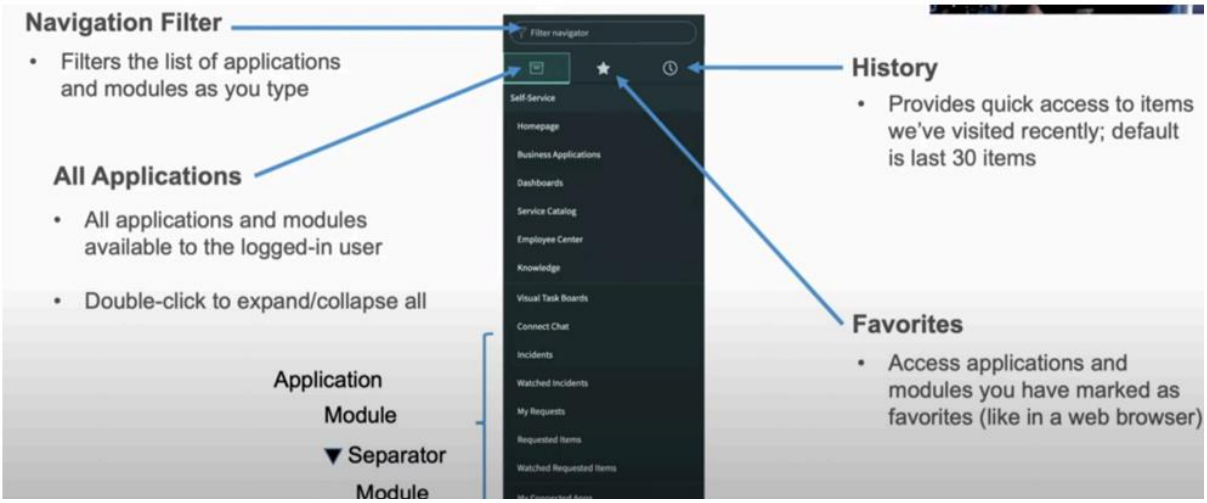
1. **Banner Frame**
2. **Application Navigator**

3. **Content Frame**

Content Frame: Displays forms, lists, dashboards, reports, and other data.

Banner Frame: Contains company logo, global search bar, notifications, user menu, and discussions sidebar (chat tool).

- **User Menu**: Allows profile viewing, impersonation, role elevation, and system settings access.
- **Global Search**: Searches the instance for records matching keywords.
- **Help**: Provides access to user guides and contextual help.



Application Navigator: Provides access to different applications and modules.

- **Favorites**: Mark frequently used applications or modules.
- **History**: Shows the last 30 accessed items.

Branding in ServiceNow

Branding within ServiceNow allows organizations to customize the look and feel of the platform to match their company's identity. This includes customizing the logo, colors, and other visual elements that appear throughout the interface. ServiceNow's branding options provide administrators with tools to ensure consistency in the user experience, which can enhance user adoption and make the platform more recognizable across teams.

Steps to Set Up Branding:

1. **Guided Setup**: Use the Guided Setup feature for a step-by-step process to apply branding settings.
2. **Logo Customization**: Replace the default ServiceNow logo with your organization's logo.
3. **Color Scheme**: Customize the platform's color scheme to match company branding.
4. **Service Portal Branding**: Modify the look of the Service Portal to provide a personalized self-service experience for users.

5. **UI Builder**: Use the UI Builder for deeper customization of the interface and branding for specific applications or modules.

Customizations like changing logos or company names can be done through system properties.

- System properties are stored in the sys_properties table.
- **Enable or disable** next experience unified navigation via the glide.polaris.next_experience property.

Tables and Columns in ServiceNow

Tables in ServiceNow are database structures that store data records. Each table holds a set of fields (columns) that define the type of information it can store. Tables can be standard (e.g., Incident, Change Request) or custom, created based on business needs. Columns in a table are equivalent to fields in a record, holding specific types of data such as text, numbers, or dates.

- **System Tables**: Built-in tables provided by ServiceNow for key functionalities such as Incident Management, Asset Management, etc.
- **Custom Tables**: Administrators can create custom tables to store specialized data for unique business processes.
- **Columns/Fields**: Define the data type (e.g., text, integer, reference) and properties like mandatory, read-only, or unique.
- **Inheritance**: Tables can inherit fields and properties from a parent table, reducing redundancy.
- **Record Storage**: Each row in the table represents a record, and columns represent attributes of that record.

Components of tables and colums:

- **System Dictionary**: Stores table and field definitions. Access via sys_dictionary or sys_db_object.
- **Sys_documentation**: Stores field labels.

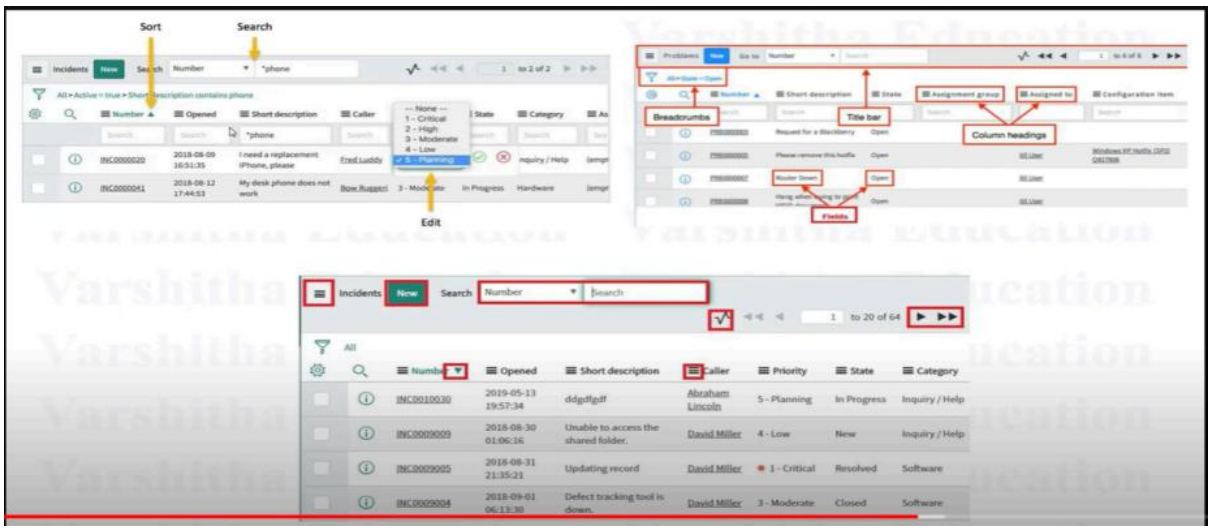
Lists in ServiceNow

Lists in serviceNow are a type of interface that displays a set of records from a table in a grid or tabular format. Lists provide a way to view, filter, sort and interact with multiple records at once.

Table_name.list is used to display the list view of a table. Table_name.LIST opens list in new table . The List Header contain many useful things to perform action on list

1. List Controls
2. Filter Lists
3. Table Search bar

4. Personalize Icon



List Views

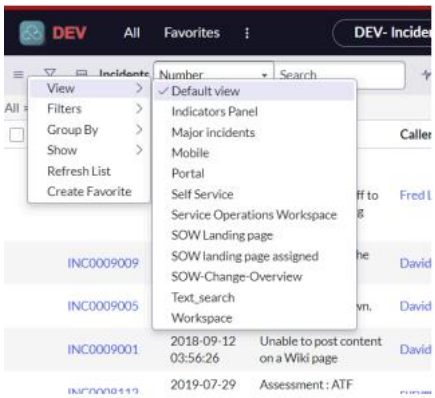
Views enable users to quickly display the same list or form in multiple ways. System administrators can create views for lists or forms.

You can create view by selecting

Control options menu - configure - List Layout - select the fields using list Collector - Scroll Down to select view - new - enter the view name – save

You can see the created view from

List Control Menu – views



Creating a classic list view

All - System definition - tables – incident

- To add new column to table - click on incident table - columns - new to add column
- to add new view - column header - column options - configure - list layout - using list collector add, remove, order the column you want in your view - list view - change from default to new - give name - save - a new list is created
- To view the new list view created - list controls - views - name of the view

List Collector

Available items that are green and followed by a plus (+) sign represent related tables To access the fields on related tables, use dot-walking

- If the column you want to add is not in the table instead in another table then use the dot- walking method - select the table in which the record is present from the list collector - click on dot walking (expand table reference fields) then you will see the column you want to add , add it

Lists display records from a table in a grid format.

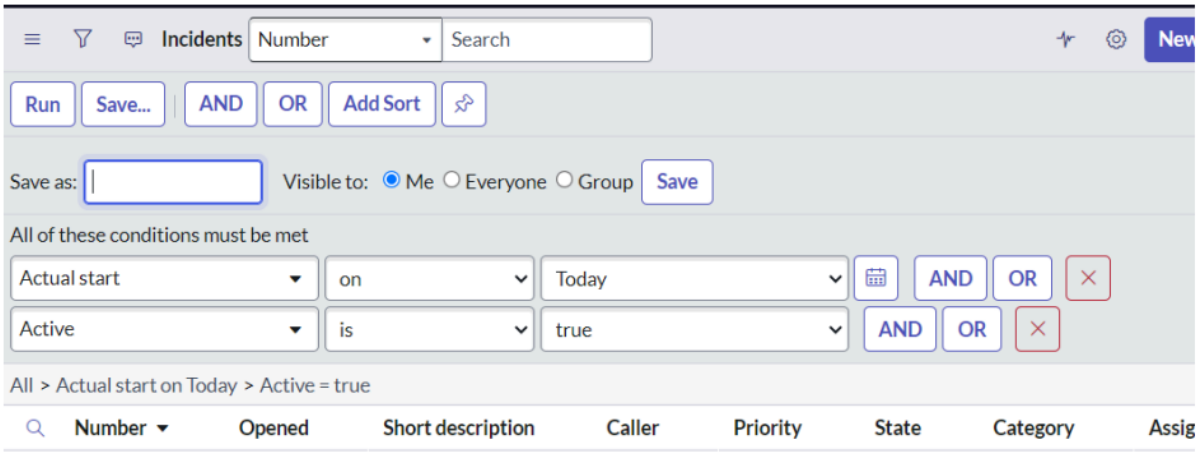
- **Table_name.list**: Opens the list view of a table.
- **Creating Views**: Modify list layout to show fields in different views. Administrators can create custom views.

Filters in ServiceNow

A filter isolates specific data from a table. It allow users to narrow down data shown in lists by applying criteria to records. They help users focus on specific information by setting conditions based on field values. Filters can be as simple as a single condition or as complex as multiple AND/OR logic combinations.

The three parts are:

1. **Field**: A choice list based on the table and user access rights. The choice list includes fields on related tables by dot-walking. Choice list based on the table and user access.
2. **Operator**: A choice list based on the field type. For example, in the incident table, the greater than operator does not apply to the Active field but it does apply to the Priority field.
3. **Value**: A text entry field or a choice list, depending on the field type.



- Filters allow users to specify criteria to view a subset of records in a list. For example, an IT technician might filter incidents based on their priority or assignment group. Users can create and save filters for easy reuse. Filters are a powerful tool for narrowing down data, helping users focus on the most relevant information.

- The ability to create, customize, and save filters is especially valuable in environments with large datasets, where it's important to quickly locate specific records.
- Add filters to your Favorites by clicking the List Controls icon and selecting Create Favorite
- In the classic list, select Run to see the results of your filter displayed in the list. To save a filter, select Save. A new field will appear where you can name your filter. After naming the filter, select who it will be visible to, then select the Save button to the right of the name and visible to options. The new filter will be available by selecting Filters from the list context menu.

Breadcrumbs

Breadcrumbs in ServiceNow visually represent the filter conditions applied to a list of records. They appear at the top of the list view and can be used to quickly refine or remove specific filter criteria. Each part of the breadcrumb corresponds to a specific filter element (field, operator, or value).

- **Modifying Filters**: Clicking a breadcrumb allows you to modify the filter without recreating the entire condition.
- **Removing Filters**: Clicking the "X" on a breadcrumb element removes that part of the filter.
- **Navigating Lists**: Breadcrumbs help users track their navigation history, showing how they arrived at a particular data view.

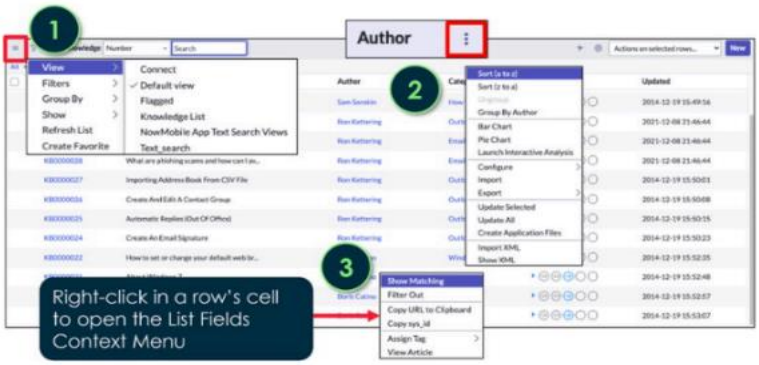
Breadcrumbs summarize filter conditions in lists. Modify conditions by selecting or removing parts of the breadcrumbs.

Context Menus in Lists

In ServiceNow, context menus in lists provide users with quick access to actions that can be performed on list items (records) or the list itself.

There are three context menus in ServiceNow lists:

1. **List Control Menu**
2. **Column Option Menu**



3. **List Field Menu**

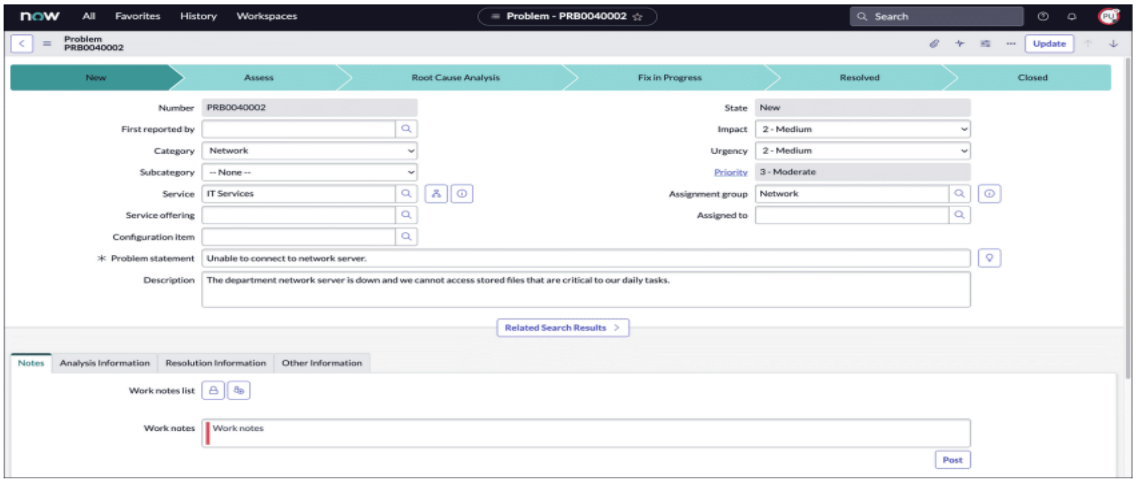
Context menus in ServiceNow lists provide additional actions that can be performed on records directly from the list view. By right-clicking on a list header or a specific record, users can access options like creating new records, exporting data, editing multiple records at once, and more.

- **List Header Context Menu**: Available actions include export, personalize list layout, and create new records.
- **Record Context Menu**: For individual records, users can perform actions such as editing, deleting, viewing details, or assigning tasks.
- **Column Context Menu**: Allows users to sort by a specific column, group records, or add/remove columns from the list view.

Forms in ServiceNow

Forms are used to display and edit individual records in ServiceNow. Forms consist of fields that capture specific information about the record. Key aspects of ServiceNow forms include:

- **Form Layout:** Forms can be customized to show the most relevant fields, sections, and tabs, ensuring that users see only the information they need.
- **Field Types:** ServiceNow supports various field types, including text, date/time, choice, reference, and more. The choice of field type depends on the type of data being captured, ensuring accuracy and consistency.
- **Form Personalization:** Users can personalize forms by rearranging fields, hiding certain sections, or adding related lists. Personalization enhances user productivity by providing a tailored experience.
- **View Creation:** Administrators can create multiple form views, each tailored for different roles or groups. For example, an IT technician might have a simplified incident form view, while a manager might have a more detailed view.



Forms are the backbone of ServiceNow's data management, providing users with an interface to interact with the data stored in the platform.
A form displays fields from one record for viewing or editing.

Configuration of the forms

- Open a list - select a record - on top left corner - additional actions - configure - list layout

Adding new fields(columns) into the form Can be done in three way:

1. Using Dictionary fields
2. Using form layout
3. Using form Design

1. **Using Dictionary field:**

   All - System definition - tables - open table - scroll down to columns - new - fill the dictionary form – save

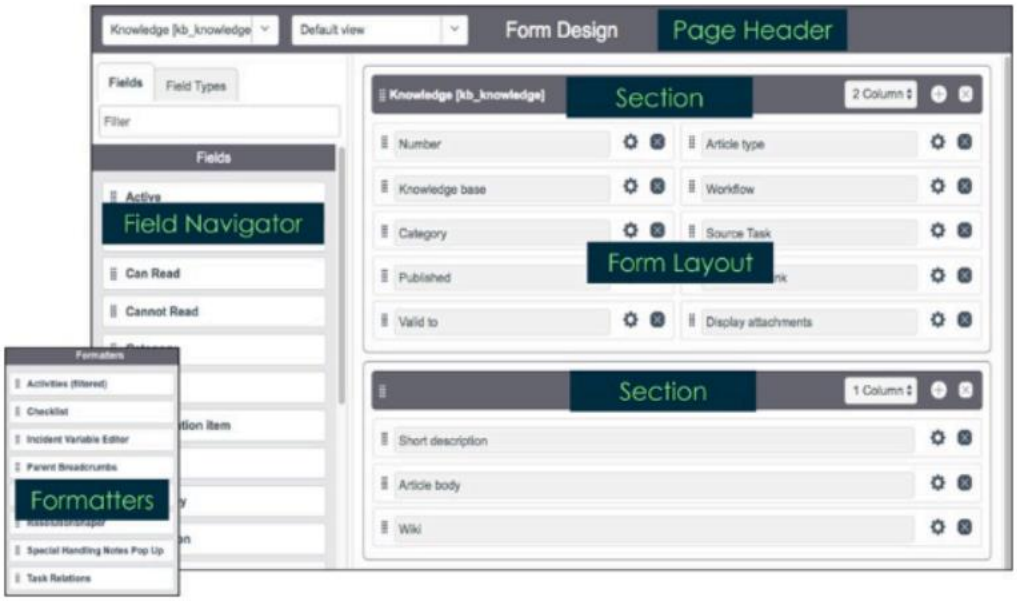
2. **Using form layout:**

   All - table_name.list (opening list of the table) - select any record - on the top left corner select additional actions (form control menu) - configure - form layout - Scroll down to create new field – add

3. **Using form Design:**

   Dot - walking does not apply here

   Can add only existing columns of the table, no new columns can be created.



Fields can be dragged and dropped to different locations on the form and new fields can be added to the form by dragging and dropping from the Fields tab or the Field Types tab. When navigating to Configure > Form Design, the Form Designer opens in a separate tab. If you try to modify the form and it turns pink, you are not in the correct application scope.

Email Notifications in ServiceNow

Notifications in ServiceNow are alerts sent to users when certain conditions are met in the platform. These can be configured to notify users about incidents, tasks, approvals, or any other system events. Notifications can be delivered through various channels, such as email, SMS, or within the platform itself.

- **Trigger Conditions**: Notifications are triggered based on specific events like record insert, update, or delete.
- **Notification Content**: Configurable to include relevant information and links to records or tasks.
- **Channels**: Notifications can be sent through email, SMS, or directly to a user's notification settings within ServiceNow.
- **Custom Notifications**: Administrators can create custom notifications for specific workflows.
- **Notification Rules**: Define when and to whom notifications are sent based on user roles, groups, or specific record conditions.

ServiceNow includes a robust notification system that keeps users informed about important events. Notifications can be triggered by specific events, such as when a task is assigned or a record is updated.

Email Notifications

Email notifications are used to send selected users email or SMS notifications about specific activities in the system, such as updates to incidents or change requests. . Types of notifications include:

1. **Email**
2. **SMS**
3. **Push Notifications**

- **Email Notifications:** ServiceNow can send automated emails based on certain conditions. These emails can include dynamic content, such as the current status of an incident or the details of a new request.
- **SMS Notifications:** For urgent communications, ServiceNow can send text messages to users' mobile devices. SMS notifications are particularly useful in incident management when immediate action is required.

- **Push Notifications:** For users of the ServiceNow mobile app, push notifications can be used to inform users of important updates or task assignments.

Notifications help ensure that users stay informed and responsive, which is critical for maintaining service levels in IT operations.

To access a new notification record

All > System Notification > Email > Notifications.

To view notifications in your instance, navigate to

All >System Mailboxes > Outbound > Outbox.

Right click on Created Date and select Preview Email.

To Creating Email Notifications

All - System Notifications - Email - Notifications - all

We have three fields to fill

1. When to send the notification
2. Whom to send the notification
3. What it will contain

When to send dropdown options are:

- Record inserted or updated
- Event is fired
- Triggered

Email Templates and Email Layouts in ServiceNow

Email templates are predefined messages that are automatically sent to users when specific events occur in ServiceNow. Email layouts determine the structure and design of these emails, allowing consistent branding and presentation of information. Together, they ensure that users receive clear and concise communications about important tasks or updates.

- **Email Templates**: Contain the body of the email, including placeholders for dynamic data from the records.
- **Email Layouts**: Define the overall design and structure, including headers, footers, and styling.
- **Personalization**: Templates and layouts can be customized to reflect an organization's branding.

- **Use Cases**: Templates are commonly used for incident updates, task assignments, approvals, and notifications.
- **Dynamic Content**: Placeholders in email templates are dynamically replaced with actual data from the ServiceNow instance when the email is sent.

Administrators can create email layouts using an inline HTML editor or manually entering HTML code.

Navigate to All > System Policy > Email > Layouts

The system stores email layout records in the Email Layout sys_email_layout table.

Creating Email templates:

1. Navigate to System Notification > Email > Templates.

2. The system displays the list of existing email templates.

3. Select the email template to which you want to apply an email layout.

4. In Email layout, select the email layout to use to format the body of email messages.

5. Click Update. The email template uses the selected email layout to format the body of email messages.

Knowledge Management

Knowledge Management in ServiceNow allows organizations to create, share, and manage knowledge articles, ensuring that users have access to critical information. This centralized knowledge base supports self-service by enabling users to find answers to common questions or issues without relying on IT support.

- **Knowledge Articles**: Documents containing information such as FAQs, troubleshooting steps, or procedures.
- **Knowledge Base**: A collection of knowledge articles categorized by topics or departments.
- **Permissions**: Access to knowledge articles can be restricted based on roles or user criteria.
- **Feedback and Ratings**: Users can provide feedback and rate articles to help improve the knowledge base content.
- **Article Versioning**: Allows maintaining different versions of an article and tracking updates over time.

ServiceNow's **Knowledge Management** application enables organizations to capture, share, and manage knowledge articles.

Knowledge Base contains Categories , Categories Contain Knowledge Articles

- To view knowledge articles
- ALL - self-service - knowledge
- ALL - Knowledge - Homepage - opens workspace containing all the Knowledge bases
- ALL - Knowledge - All - open table containing all the Knowledge Articles

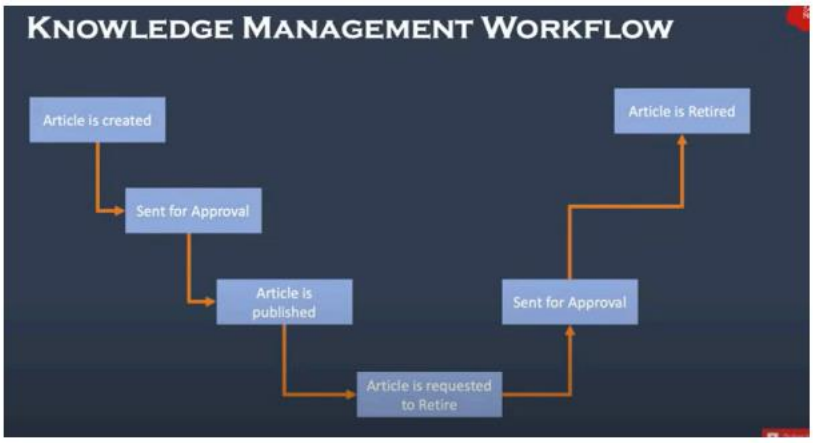
We have to have a role of Knowledge , Knowledge_admin, Knowledge_manager to access KA

Key features include:

- **Knowledge Bases:** These are repositories where organizations can store articles, procedures, and best practices. Knowledge bases are organized by categories to make it easier for users to find the information they need.
- **Article Creation and Approval:** Users can create knowledge articles, which must go through an approval process before being published. This ensures that the information is accurate and up-to-date.
- **Search Capabilities:** The platform includes powerful search tools, allowing users to search for articles based on keywords or filters. This is particularly useful for support agents looking for troubleshooting guides or procedures.

Creation of New Knowledge articles the cycle is

1. Draft a article
2. Sent for approval Manager
3. Publish the article
4. Get feedback and rating
5. Retire the article
6. Sent for Approval
7. Article is retired



The Knowledge homepage displays knowledge articles organized by Knowledge Base and Category. An article can only be associated with one knowledge base.

From the homepage, users with the correct permissions can import a Word document to a Knowledge Base using the Import Articles button and create a new article by clicking Create an Article.

Creating an article

- All - Knowledge - all - new - fill the form - Click on publish - Approve request is sent

Approval of the Article Publish

- Impersonate Approver - all - servicedesk - my approvals - open record - approve
- As system administrator you can open Knowledge article records - scroll to related lists
  Approvals - Approve .

User Criteria in Knowledge Management

User Criteria determine which users can create, read, or write knowledge articles. Outcomes include:

- **canRead**: Users can read all articles.
- **cantRead**: Users cannot read, create, or modify articles.
- **canContribute**: Users can read, create, and modify articles.

User Criteria in Knowledge Management

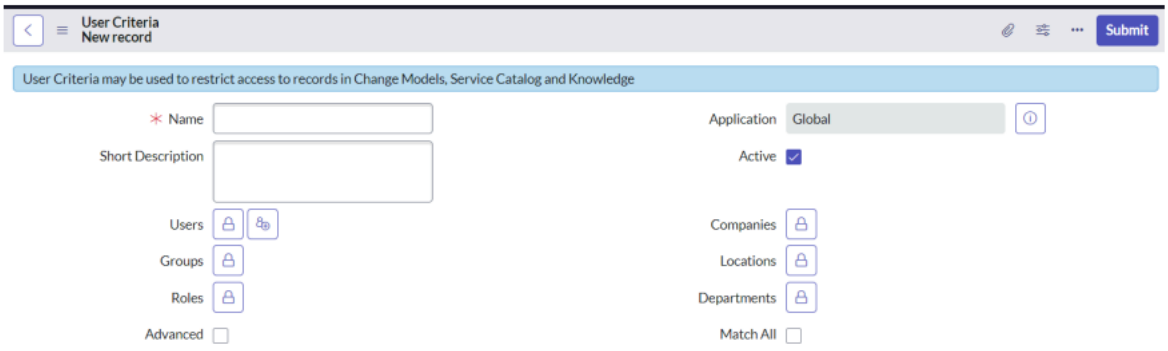
User Criteria in Knowledge Management allow administrators to control who can view, contribute, or manage knowledge articles in ServiceNow. Criteria can be based on roles, groups, or specific attributes like department or location. This ensures that users only have access to relevant articles while preventing unauthorized access to sensitive information.

- **View Criteria**: Determines who can view articles based on predefined user attributes.

- **Contribute Criteria**: Specifies which users or groups can create or edit knowledge articles.
- **Manage Criteria**: Defines which users have administrative rights over the knowledge base, including article publishing and review.
- **Dynamic Access**: User criteria can be dynamically adjusted to reflect changes in roles or organizational structure.
- **Application of Criteria**: Criteria can be applied globally to an entire knowledge base or to individual articles.

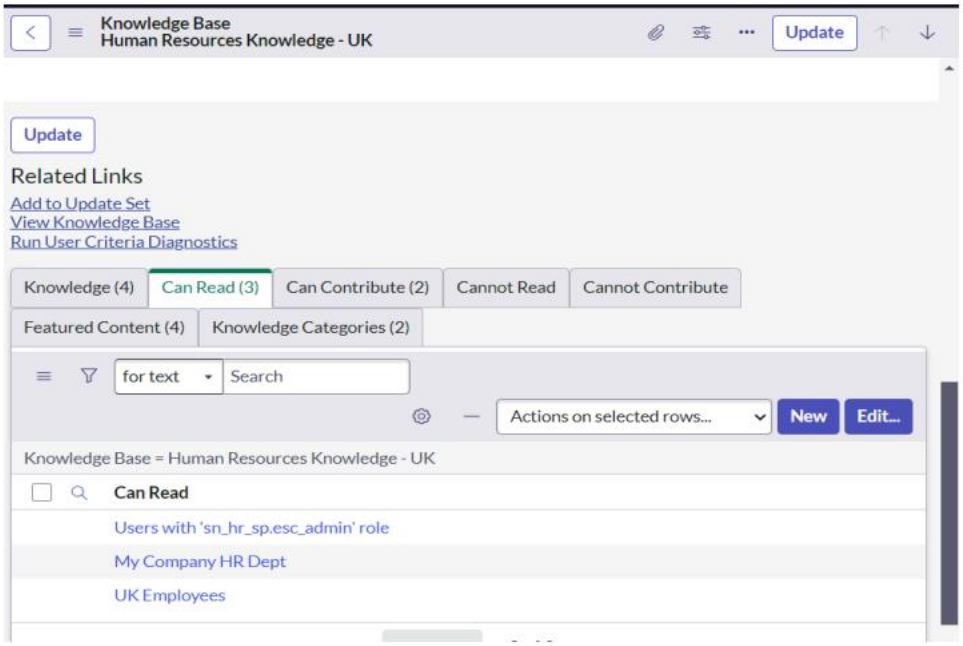
To create a User criteria

- All - knowledge - User criteria – New



To implement user criteria, navigate to

All > Knowledge > Knowledge Bases and select a knowledge base - User Criteria records are accessed from the Can read or Can contribute related lists



Service Catalog in ServiceNow

The Service Catalog in ServiceNow is a user-friendly interface that provides a list of available services and products within an organization. Users can request hardware, software, or other services, and the platform automates the approval and fulfillment processes. The Service Catalog simplifies self-service, enabling users to find and request items without IT intervention.

- **Catalog Items**: Individual services or products available for request.
- **Categories**: Items are organized into categories for easier navigation.
- **Workflows**: Service Catalog requests trigger predefined workflows for approval and fulfillment.
- **Approval Management**: Requests can be routed to managers or departments for approval.
- **Service Levels**: SLAs (Service Level Agreements) are used to track the time taken to fulfill requests.

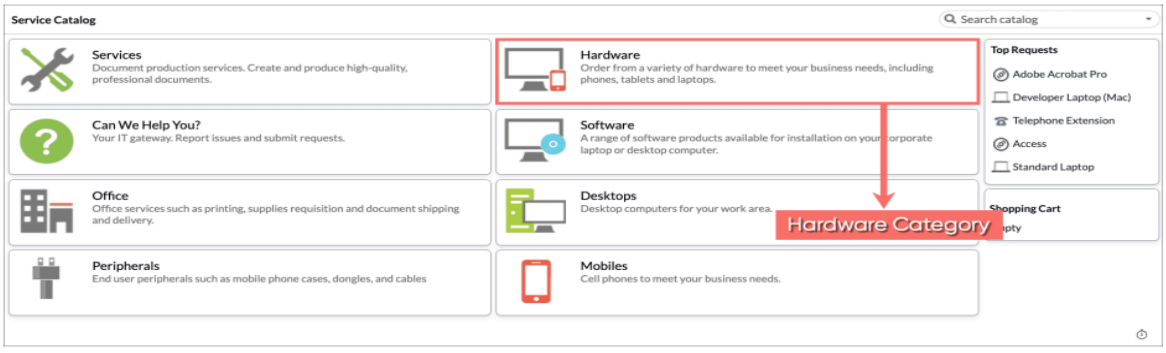
The Service Catalog allows users to request products and services.

To create a new item or modify an existing item, navigate to

- Go to ALL > Service Catalog > Catalog Definitions > Maintain Items.

The **Service Catalog** provides a self-service portal for users to request services, such as hardware provisioning, software installation, or access to applications. Features of the Service Catalog include:

- **Catalog Items:** These are individual services or products that users can request. Catalog items can include forms that capture the necessary information for fulfilling the request.
- **Workflows:** Each catalog item can have an associated workflow, automating the request approval process, fulfillment, and notification of completion. This reduces manual intervention and speeds up the delivery of services.
- **Categories and Subcategories:** The catalog is organized into categories, making it easier for users to browse and find the services they need. For example, there might be categories for IT services, HR services, and Facilities services.



The Service Catalog improves user satisfaction by providing a centralized portal where users can quickly and easily request the services they need, while automation ensures that these requests are processed efficiently.

Variables and Variable Sets in ServiceNow

Variables in ServiceNow allow users to provide specific input when submitting a Service Catalog request. Variable Sets are groups of variables that can be reused across multiple catalog items, making it easier to manage related data inputs across different requests.

- **Variables**: Fields that capture user input, such as text, choice, or reference fields.
- **Variable Sets**: A collection of variables that can be reused across multiple catalog items.
- **Variable Types**: Text, date, choice, and reference fields are some of the variable types available.
- **Dynamic Variables**: Change based on user input, providing a personalized experience.
- **Dependent Variables**: Variables whose values depend on the selection of other variables in the form.

To create a new variable set

Navigate to All > Service Catalog > Catalog Variables > Variable Set

Variables define the structure of a catalog item form. Common variable types include:

- **Multiple Choice**: Radio buttons for predefined options.
- **Select Box**: A dropdown list.
- **Single Line Text**: Single-line text field.
- **Reference**: Specifies a record in another table.

Record Producers

Record Producers are a type of Service Catalog item that allows users to create records in any table within ServiceNow. Unlike standard catalog items, which typically result in a request or task, record producers create records directly in tables such as Incident, Change, or Custom tables.

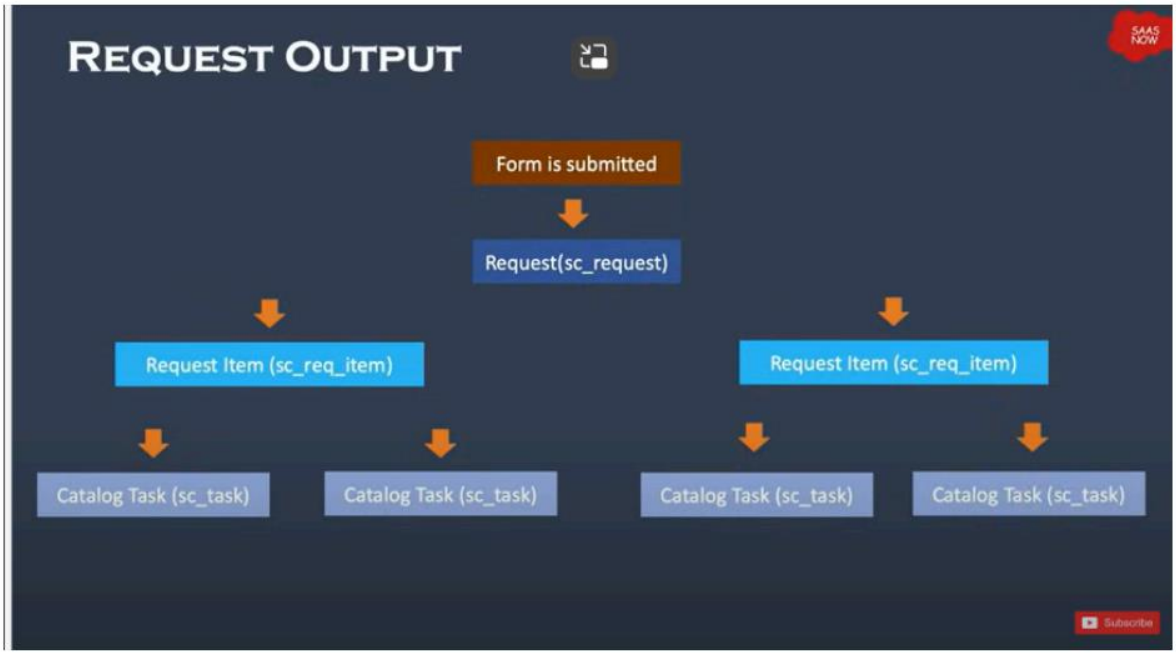
- **Purpose**: Streamlines the creation of records by providing a simple form for users to complete.
- **Customization**: Forms and fields in record producers can be customized based on the business process.
- **Trigger Workflows**: Record producers can trigger workflows or approvals automatically once the record is created.

- **Data Mapping**: Field values entered by the user are mapped to corresponding fields in the table where the record is created.
- **Reusable**: Record producers can be reused in different catalog items

Record Producers allow users to create records like incidents from the service catalog. They simplify complex tasks for end users.

Service Catalog Item Request Output

For Catalog items, a request is created. A request can have one or more items associated with it. An item can have one or more tasks associated with it. Each output is stored in the appropriate corresponding table.



REQ# Request [sc_request] table: A request number generated to keep track of an order.

Records on this table begin with REQ and behave like containers.. REQ record is the shopping

cart. It can contain one or many items.

RITM# Requested Item [sc_req_item] table: Records on this table begin with RITM and manage

the delivery of each individual item in the request. Within a request generated from a catalog order, each discrete item ordered is given a specific “Requested Item Number” known as an RITM (number).

SCTASK# Catalog Task [sc_task] table: Records on this table begin with SCTASK and are the

assigned tasks needed to complete the delivery of each Request item from start to finish. Some

of the more important fields are the Assignment group, the Due date, Work start, and Work end

dates.

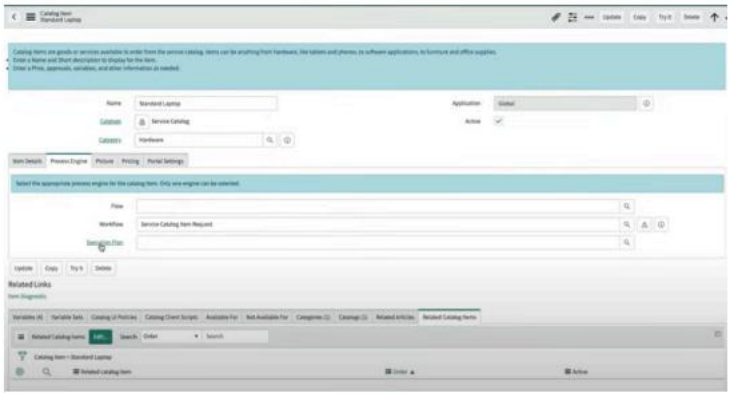
Catalog Builder - It is used to build Custom Catalog Items

Process Stages

Flow stages attached to an item indicate the progress or state of an item in the delivery process

with one of the following stages:

1. Waiting for approval (In Progress)
2. Approved
3. Pending (has not started)
4. Fulfillment (In Progress)
5. Deployment/Delivery
6. Completed



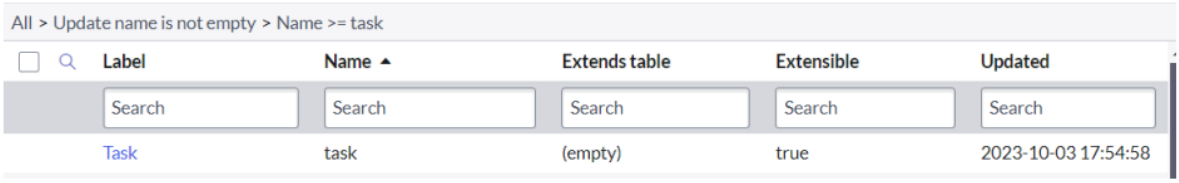
After a request has been submitted, users are able to easily track it by navigating to

All > Self-Service > My Requests

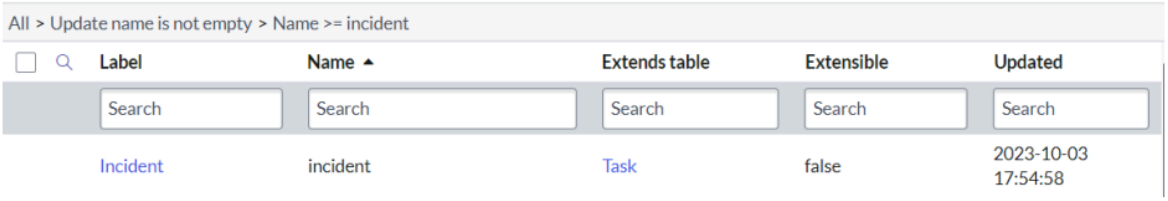
and opening the record associated with the request.

Properties of Table in ServiceNow

In ServiceNow, tables store data in rows and columns, where each row is a record, and each column represents a field. The **properties of a table** define how data is managed and interacted with. Here's a detailed breakdown:



- **Label**: The name displayed to users when they view the table, typically user-friendly. This is what appears in lists, forms, and drop-down menus.
- **Name**: This is the unique system name of the table. Custom tables are usually prefixed with "u_". For example, u_custom_table is a custom table.
- **Table Space**: Each table has an allocated space in the database that stores the records. Monitoring table space is crucial for system performance, especially for tables that store large volumes of data.
- **Fields**: Tables consist of columns known as fields, each with its own data type (e.g., String, Integer, Date). Field properties include mandatory status, default value, and field length.
- **Data Dictionary**: This defines field types, character limits, and default values across the table.
- **Indexes**: Tables can have indexes for faster query performance. Indexes are created on key fields to enhance the efficiency of searching and filtering data.
- **Access Control**: Table-level security settings determine who can access and manipulate the records. Roles and ACLs (Access Control Lists) are applied to ensure secure data management.



Example: The incident table has properties such as a label (Incident), name (incident), and fields like **State**, **Priority**, and **Assigned To**. It also has security policies ensuring that only authorized users can view or modify specific incidents.

Schema Map

The **Schema Map** provides a visual representation of a table's relationship with other tables. It helps developers and administrators understand data structure and table interactions. The schema map includes various relationships like **references** and **extensions**, offering a clear overview of the data model.

Key Components:

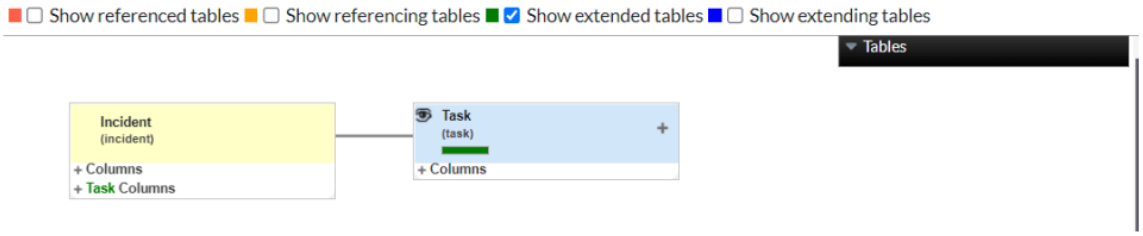
- **Parent Table**: Shows tables that the current table extends from. For instance, the incident table extends from the task table.
- **Referenced Tables**: Shows which other tables are referenced within the current table. For example, the incident table references the user table for fields like **Opened By** or **Assigned To**.
- **Extended Tables**: Displays tables that extend from the current table. These tables inherit fields from the parent table but can also include additional fields.

Benefits:

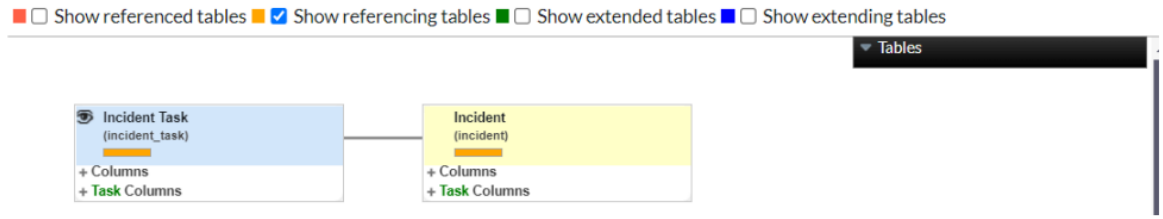
- **Optimization**: Helps optimize queries by understanding relationships and dependencies.
- **Troubleshooting**: Aids in identifying and resolving issues by visualizing table dependencies.
- **Data Flow Analysis**: Helps in mapping out how data moves across tables, useful for designing workflows or integrations.

Schema Map of Incident Table

The **Incident table** is one of the most critical tables in ServiceNow's IT Service Management (ITSM) system. It is used to log and manage incidents that disrupt normal business operations. The schema map for the incident table is highly interconnected with several other tables.



Referencing tables is incident task because it contains a field that refers to incident records



Key Relationships in Incident Table Schema Map:

- **Task Table**: The incident table extends the task table, meaning it inherits all fields from the task table, such as **State**, **Priority**, and **Assigned To**.

- **User Table**: The incident table references the sys_user table for fields like **Opened By**, **Caller**, and **Assigned To**.
- **Group Table**: For group assignments, the incident table references the sys_user_group table, which holds information on user groups.
- **CMDB (Configuration Management Database)**: The cmdb_ci table is referenced in incidents to track the affected configuration items (CIs). This is critical for understanding which assets are impacted by the incident.

Practical Example: In the incident schema map, you may see that the incident table pulls data from the sys_user table when assigning tickets. Understanding this schema is crucial for customizing workflows, automation, and reports related to incident management.

Types of Table Relationships

ServiceNow supports various types of table relationships, each defining how tables interact with each other. Understanding these relationships is crucial for data modeling and system customization.

1. **One-to-One Relationship**:
    - In this relationship, one record in a table corresponds to exactly one record in another table.
    - **Example**: A user's profile in sys_user can have a one-to-one relationship with their employee record in a custom employee_details table.
2. **One-to-Many Relationship**:
    - This is a common relationship where one record in a table can be related to multiple records in another table.
    - **Example**: A single user (from sys_user) can be assigned multiple incidents (from incident).
3. **Many-to-Many Relationship**:
    - This relationship allows multiple records in one table to be linked to multiple records in another through an intermediary table.
    - **Example**: Users and groups have a many-to-many relationship through the sys_user_grmember table. A user can belong to multiple groups, and a group can have multiple users.
4. **Extensions**:

- Tables can extend other tables, inheriting all fields from the parent table while allowing new fields to be added.
- **Example**: The incident table extends the task table, inheriting all task-related fields while adding incident-specific fields like **Caller** and **Category**.

Use Cases:

- **One-to-Many** is used in task assignment scenarios, where one user manages many tasks.
- **Many-to-Many** relationships are ideal for managing complex role-based access scenarios, where multiple users share responsibilities across departments.

Extensions

Extensions are a key feature in ServiceNow that allow tables to inherit fields and logic from a parent table, promoting reuse and standardization. When a table extends another, it inherits all fields and business logic from the parent.

Features of Extensions:

- **Field Inheritance**: The child table automatically inherits all fields from the parent table but can also define additional fields.
- **Business Logic Inheritance**: Business rules, UI policies, and workflows applied to the parent table can also apply to the child table unless explicitly overridden.
- **Reusability**: Extensions promote reusability of data models. For example, the incident table inherits fields like **Priority**, **State**, and **Assignment Group** from the task table, while adding additional fields specific to incident management.

Benefits:

- Reduces duplication of data and logic.
- Ensures consistency in workflows.
- Simplifies the process of extending the system with new tables.

Example: The change_request and problem tables also extend the task table, inheriting common fields such as **Short Description**, **State**, and **Assigned To**, but introducing their own fields specific to change and problem management.

Access Control List (ACL)

The **Access Control List (ACL)** is a fundamental part of ServiceNow's security model. It defines the permissions required to access and interact with data at various levels—table, record, and field.

Key Components of ACLs:

- **Roles**: ACLs are often role-based. Only users with specific roles can view, create, update, or delete records.

- **Conditions**: You can set conditions within an ACL to apply more specific rules. For instance, only incidents in a particular state (like "New") may be viewable by certain roles.
- **Scripted ACLs**: Advanced ACLs can include JavaScript-based conditions, offering dynamic access control based on complex logic.

Types of ACL Rules:

1. **Table-level ACLs**: Control access to entire tables.
2. **Field-level ACLs**: Control access to individual fields within a table. For example, a user might be able to view a record but not a sensitive field like **Cost**.
3. **Row-level ACLs**: Control access to specific records within a table based on conditions, such as whether the user is assigned to the incident.

Example: An ACL can be created to ensure that only users with the **itil_admin** role can delete incidents, while regular IT users can only update or view incidents.

Access Control: Row and Column Level

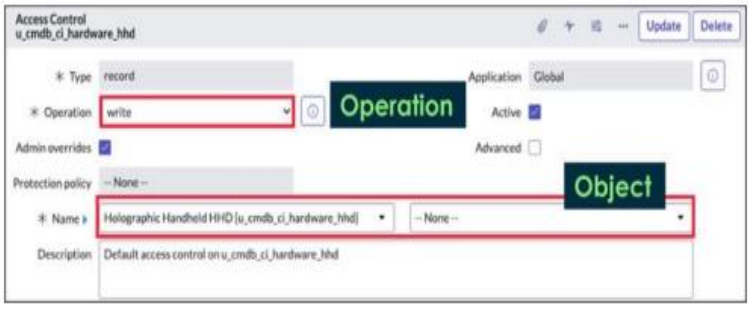
Access control in ServiceNow operates at both the **row** (record) and **column** (field) level, providing fine-grained control over data security.

- **Row-level Access Control**: Determines whether a user can access an individual record. For instance, you can restrict access to only those records where the **Assigned To** field matches the current user.
- **Column-level Access Control**: Controls access to specific fields within a record. For example, while a user might be able to view an incident, certain fields like **Impact** or **Assignment Group** might be hidden based on their role.

Importance:

- Protects sensitive information by controlling who can see or edit particular fields.
- Ensures that users only interact with data that is relevant to their role or responsibilities.
- Allows for complex security models where access is tailored to specific user roles or organizational units.

Example: An HR employee may have access to view an employee record (row-level access), but not see sensitive fields like **Salary** or **Social Security Number** (column-level access).



Viewing ACL

Viewing Access Control List (ACL) rules in ServiceNow helps administrators and developers understand which security policies apply to different tables, records, and fields. Here's how to view and interpret ACLs:

Steps to View ACLs:

1. **Navigate to the ACL Configuration**:
    - Go to **System Security** > **Access Control (ACL)** in the Application Navigator. This will list all ACLs in the system.
2. **Filtering ACLs**:
    - You can filter ACLs based on the table, operation (create, read, write, delete), or specific fields. For instance, to view ACLs for the incident table, apply a filter to see all related rules.
3. **Understanding ACL Structure**:
    - Each ACL is defined with specific parameters such as **Operation**, **Type**, and **Condition**. For instance:
        - **Operation**: What action the ACL governs (e.g., create, read, write).
        - **Type**: Whether the ACL applies to a table or a specific field.
        - **Condition**: The conditions that must be met for the ACL to take effect.

Interpreting ACLs:

- **Type**: Identifies whether the ACL applies to an entire table or just a field.
- **Operation**: Determines what kind of access (read, write, delete) is controlled by the rule.
- **Script/Conditions**: Advanced ACLs may contain JavaScript to check conditions dynamically. For instance, an ACL script may allow only certain users to update incidents that are in a specific state.

Example:

In an incident table, you might find multiple ACLs:

- A **read** ACL for the whole table that permits all users to view incident records.
- A **write** ACL for the **State** field that only allows users with the **itil_admin** role to change the state of an incident.

Key Insights:

- Viewing ACLs is critical for debugging access issues, especially when users cannot perform expected actions like editing a field.
- ACLs can be layered, meaning that more than one rule may apply to the same object.

Access Control: Rule Types

Access Control List (ACL) **rule types** in ServiceNow determine how permissions are applied at different levels of the platform. These rules enforce security at the table, record, and field level, providing fine-grained control over who can access and modify data.

1 2 3

Common ACL Rule Types:

1. **Table Rules**:
    - These ACLs apply to an entire table. If a user has access to the table, they can interact with records based on further conditions. For instance, a user may have permission to read records from the incident table but not edit them.
2. **Field Rules**:
    - These ACLs apply to individual fields within a record. For example, an IT technician may be able to view and edit an incident but may not have permission to view sensitive fields like **Caller** or **Resolution Code**.
3. **Record Rules**:
    - These ACLs determine access to specific records within a table. For example, a technician may be allowed to see only incidents assigned to them, controlled by a row-level ACL.
4. **Conditional Rules**:
    - These ACLs use scripts to evaluate specific conditions. For example, only users in a particular department may be allowed to view certain records, or only incidents in certain states can be edited by a user.

5. **Scripted ACLs**:
    - Allows the use of JavaScript in ACL rules to dynamically control access based on complex logic. For instance, an ACL can be scripted to grant access only if the current user is the manager of the record's **Assigned To** user.

Importance:

- Different rule types allow granular control over who can see or modify data.
- Scripted ACLs provide flexibility for complex business logic.

Example: A **field ACL** might prevent users from viewing the **Salary** field on an employee record, while a **table ACL** controls overall access to the employee table.

Creating Access Control

Creating ACL rules in ServiceNow allows administrators to define who can access or modify specific tables, fields, or records. Here's how to create an ACL:

Steps to Create an ACL:

1. **Navigate to ACL Creation**:
    - Go to **System Security** > **Access Control (ACL)** > **New**.
2. **Define the ACL Properties**:
    - **Type**: Choose whether the ACL applies to a table, record, or field.
    - **Operation**: Define the action that this ACL controls (e.g., create, read, write, delete).
    - **Table**: Specify the table or field the ACL applies to.
    - **Role/Condition**: Select which roles have access or create a scripted condition to control access dynamically.
3. **Writing Conditions or Scripts**:
    - You can set conditions using the condition builder, or use a JavaScript script to evaluate more complex scenarios.
4. **Save and Test**:
    - After creating the ACL, test it by impersonating different users to ensure that permissions are applied correctly.

Example:

Let's say you want to create an ACL that only allows users with the **itil_admin** role to delete incidents:

- **Type**: Table
- **Table**: Incident

- **Operation**: Delete
- **Condition**: Role = itil_admin

Elevate Role

Elevating roles in ServiceNow allows administrators or privileged users to temporarily increase their permissions to perform tasks that require higher-level roles, such as modifying access controls or changing critical system configurations.

Steps to Elevate Role:

1. **Elevate Role Feature**:
    - In the top right corner of the ServiceNow interface, click on the user menu and choose **Elevate Role**.
2. **Choose the Role**:
    - A list of available elevated roles appears. The most common elevated role is **security_admin**, which allows you to modify security settings.
3. **Perform the Task**:
    - Once you elevate the role, you can perform the necessary administrative actions, such as editing ACLs or changing system properties.
4. **Role Expiration**:
    - Elevated roles are temporary and will expire after a set period or after logging out. This ensures that elevated permissions do not remain active longer than necessary.

Example:

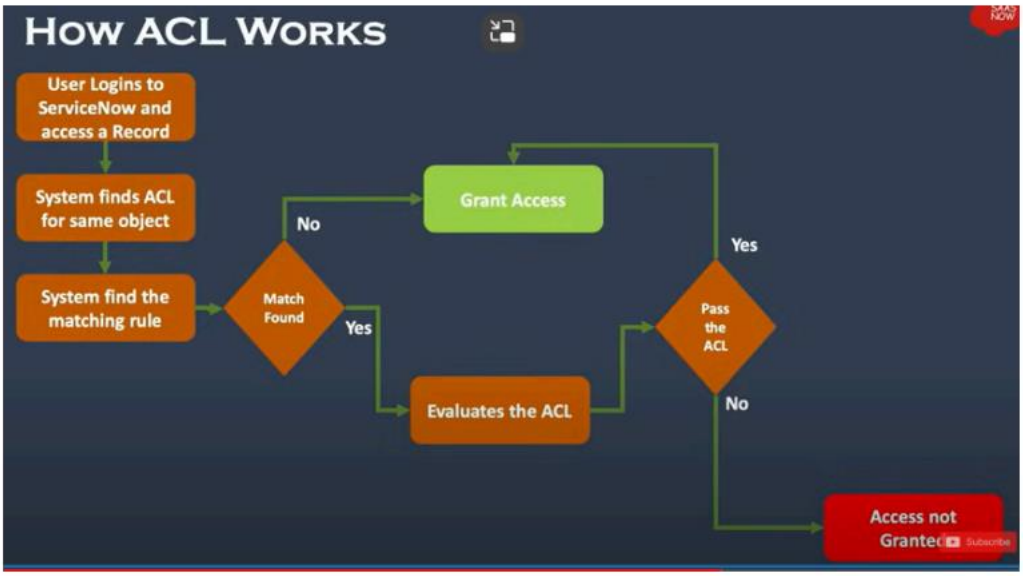
An administrator elevates to the **security_admin** role to create new ACLs for restricting access to sensitive fields in the **incident** table.

Security Implications:

- Elevating roles is a powerful feature, but it must be controlled carefully. Only trusted users should have the ability to elevate roles, as this grants temporary access to critical security settings.

Evaluating ACL

Evaluating ACLs involves understanding how ServiceNow processes multiple ACL rules to determine whether a user can access a particular record or field. ServiceNow evaluates ACLs in a specific order to ensure that the most specific ACL takes precedence.



How ACL Evaluation Works:

1. **Match on Table/Field**: First, ServiceNow identifies which ACLs apply to the requested table or field.
2. **Operations Check**: It checks the operation (e.g., read, write, delete) to see if the user has permission for that specific action.
3. **Role Verification**: ServiceNow verifies whether the user's role matches the ACL requirements.
4. **Condition Evaluation**: If the ACL has additional conditions (like only users in a certain department), these conditions are evaluated next.
5. **Script Execution**: Finally, any scripted conditions in the ACL are executed to dynamically determine access.

Example:

If a user tries to access a field in the **incident** table, ServiceNow evaluates the table-level ACLs first, then checks for any field-level ACLs that might apply. If the field-level ACL denies access, the user cannot view or edit that field, even if they have permission to see the record as a whole.

Best Practices for Evaluating ACLs:

- Use the **Security Diagnostics** tool to troubleshoot ACL issues.
- Avoid unnecessary complexity in ACL conditions to maintain system performance.

Import Sets in ServiceNow

Import Sets in ServiceNow allow administrators to import data from external sources (such as Excel, CSV, or database connections) into ServiceNow tables. Import sets provide a flexible method to move data into the platform for configuration management, asset tracking, or custom applications.

Key Concepts:

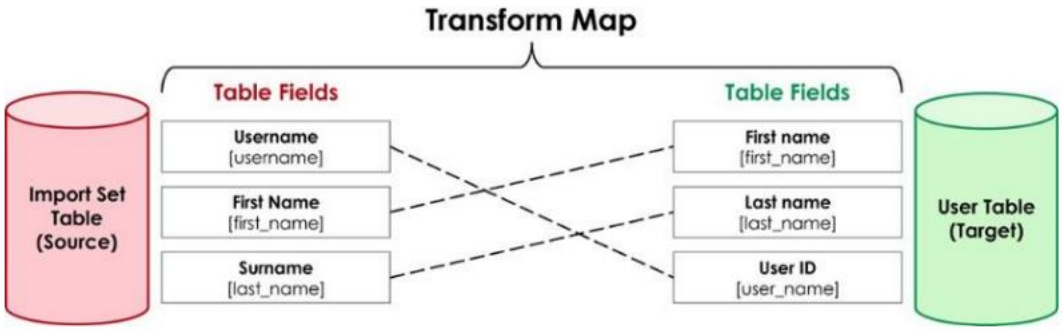
- **Import Set Table**: A temporary table that stores imported data before it is transformed into the target table.
- **Transform Map**: Defines how data from the import set is mapped to fields in the target table.
- **Data Source**: The location from which the data is imported (e.g., file, FTP, or web service).

Steps to Use Import Sets:

1. **Create a Data Source**: Define where the data is coming from (e.g., upload an Excel file).
2. **Create an Import Set Table**: This is a staging area for the imported data. It mirrors the structure of the source data.
3. **Create a Transform Map**: Map the fields from the import set table to the fields in the target table (e.g., mapping an Excel column to a **User** table field).
4. **Run the Import**: Import the data into the target table by running the transform map.

Transform Maps

Transform maps in ServiceNow are crucial for defining how data imported via **Import Sets** is mapped to the corresponding fields in the target tables. They serve as a bridge between the staging area (Import Set Table) and the target table, ensuring that data is transferred correctly and efficiently.



Key Elements of a Transform Map:

1. **Source Table**: The Import Set table where the data is initially stored. This table mirrors the structure of the external data source (such as a CSV or Excel file).
2. **Target Table**: The table where the data will eventually be imported. This is typically a core ServiceNow table like **User**, **Incident**, or a custom table.
3. **Field Mapping**: Defines which fields in the source table should map to which fields in the target table. For example, a **Name** column in the import set might map to the **Full Name** field in the target table.

Steps to Create a Transform Map:

1. **Create the Transform Map**:
    - Navigate to **System Import Sets** > **Create Transform Map**.
    - Specify the **Source Table** (Import Set table) and the **Target Table** where the data should go.
2. **Map Fields**:
    - Define the mapping between source table fields and target table fields. This can be done manually or by using an auto-map feature that suggests field mappings based on similar field names.
3. **Scripts and Conditions**:
    - Transform maps can include **Before/After** transform scripts, which allow for complex data manipulation during the transformation process. For instance, a script could format names, calculate values, or even create related records based on the imported data.
4. **Run Transform**:
    - After the mapping is done, run the transform to move the data from the import set table to the target table.

Example:

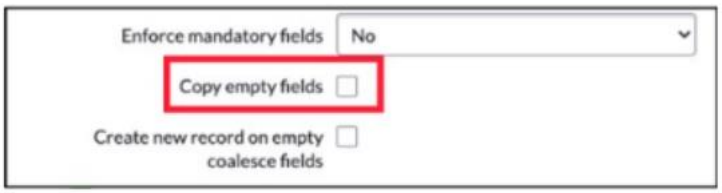
Suppose you are importing employee data. The **Transform Map** maps fields from an external Excel sheet (Import Set table) like **First Name**, **Last Name**, and **Email** to the corresponding fields in the **User** table in ServiceNow.

Key Insights:

- Transform Maps can be reused for similar data imports, making them highly efficient for recurring data migrations.
- They support **on-the-fly transformations**, such as data validation, cleaning, or formatting, ensuring data quality in the target table.

Coalesce Fields

Coalesce fields play a critical role in defining how ServiceNow matches incoming data to existing records when importing via an Import Set. The coalesce field acts as a unique identifier to determine if a record should be updated or inserted as new.



How Coalesce Works:

1. **Single Coalesce Field**:
    - If a single field is marked as coalesce, ServiceNow will attempt to find an existing record in the target table with the same value in that field. If it finds a match, the existing record is updated. If no match is found, a new record is created.
2. **Multiple Coalesce Fields**:
    - If multiple fields are coalesced, ServiceNow requires that all coalesced fields match for an update to occur. Otherwise, a new record is inserted.
3. **No Coalesce Field**:
    - If no field is marked as coalesce, all imported data is treated as new records, and no updates are made to existing records.

Example:

In a **User** table import, if you mark **Email** as the coalesce field, ServiceNow will look for existing records with the same email address. If a record is found, it will update the other fields (e.g., First Name, Last Name) in that record. If no matching email is found, a new user record is created.

Best Practices:

- Use a field that is guaranteed to be unique (e.g., Email, Employee ID) as the coalesce field to avoid duplicate records.
- Ensure that the data in the coalesce field is clean and consistent to prevent update errors.

Two Types of Mappings in Transform Map

There are two primary types of mappings done in a **Transform Map** in ServiceNow:

1. Direct Field Mapping:

- This is the most straightforward type of mapping, where fields in the **Import Set Table** are directly mapped to fields in the **Target Table**.
- For example, mapping an **Email** field in the import set to the **Email** field in the **User** table.

2. Scripted Mapping:

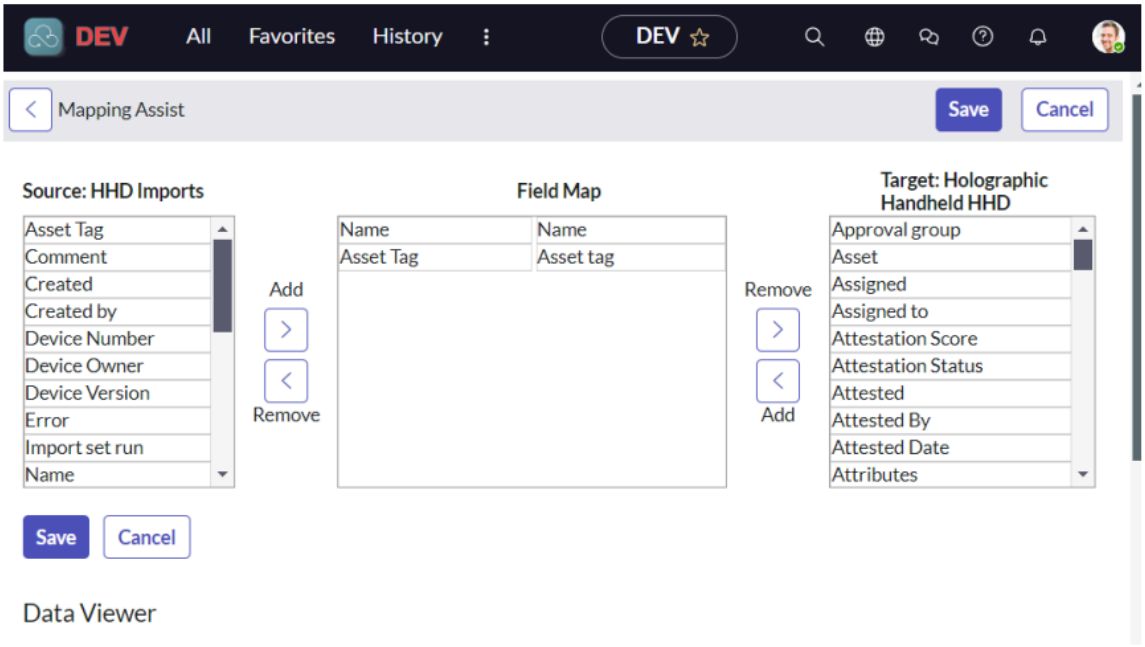
- In cases where data needs to be transformed or manipulated before it is mapped, **scripted mapping** is used.
- For instance, if the data coming from the import file has a different format (e.g., a full name field that needs to be split into first and last names), JavaScript can be written to manipulate the data before it's mapped to the target fields.

Key Points:

- Direct mapping is simpler and faster but assumes that the data format in the source matches that of the target.
- Scripted mapping provides flexibility, allowing data transformations but requires more setup and testing.

Process to Import Data into ServiceNow Table from Excel

Importing data from Excel into ServiceNow involves using Import Sets and Transform Maps to ensure data is accurately brought into the platform. Here's the step-by-step process:



Steps to Import Data:

1. **Create a Data Source**:
    - Navigate to **System Import Sets** > **Create Data Source**.
    - Upload the Excel file as the source. Define the format (e.g., Excel, CSV) and ensure that the file columns match the fields you want to import.
2. **Create Import Set Table**:
    - The Import Set Table stores the raw data temporarily before transformation.
3. **Create a Transform Map**:
    - Map the fields from the Import Set Table to the corresponding fields in the target table (such as Incident, User, or a custom table).
4. **Run the Import**:
    - Execute the import process. The data from Excel is first stored in the Import Set Table, and then the Transform Map is used to move the data to the target table.
5. **Verify the Import**:
    - After running the import, verify the data by inspecting the target table to ensure that the data was accurately imported.

Best Practices:

- Validate the data in Excel before importing it to ensure that it adheres to the target table's field formats.

- Use coalesce fields to prevent duplicate records.

CMDB (Configuration Management Database)

The **Configuration Management Database (CMDB)** in ServiceNow is a repository that acts as a data warehouse for IT installations. It is used to store information about hardware, software, and other assets (known as Configuration Items or CIs) that are crucial to IT services.

Key Functions of the CMDB:

- **Centralized Repository**: The CMDB stores and manages data about IT infrastructure, from physical servers to virtual machines, and even software components.
- **Service Mapping**: The CMDB maps services to the IT infrastructure, helping teams visualize how services depend on different hardware and software.
- **Relationship Management**: It tracks relationships between CIs, such as which servers host specific applications, or which applications are required to run a business service.

Example Use Cases:

- **Incident Management**: When an incident occurs, the CMDB can help quickly identify which systems are affected based on relationships between CIs.
- **Change Management**: Before making changes to a service, the CMDB can show how changes might impact other systems or services.

CMDB is vital for ensuring service reliability, troubleshooting, and risk management by providing a clear view of the IT environment.

Key Tables of CMDB

In ServiceNow, the CMDB uses several key tables to store and manage configuration data:

1. **cmdb_ci**:
    - This is the base table for all configuration items (CIs). All configuration items extend this table, meaning that fields and relationships common to all CIs are stored here.
2. **cmdb_rel_ci**:
    - This table tracks relationships between different CIs. For instance, it shows which servers are connected to specific applications or networks.
3. **cmdb_ci_computer**:

- This table specifically stores information about computing devices (e.g., servers, desktops, laptops). It inherits fields from **cmdb_ci** but adds more specific fields like CPU, RAM, and disk space.

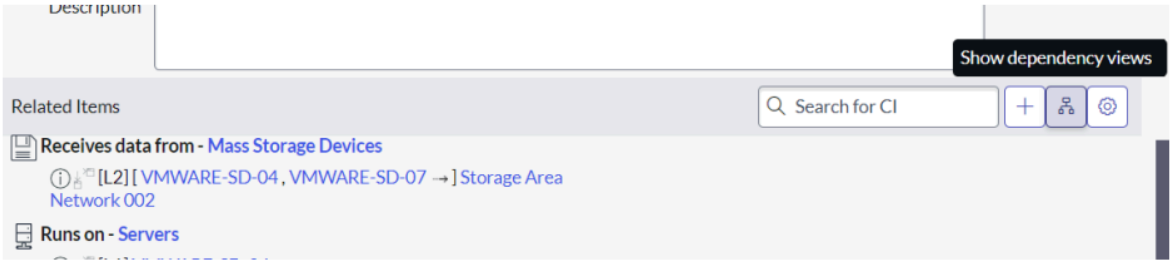
4. **cmdb_ci_service**:
    - This table is used to store data about services in the organization (e.g., email services, database services). It is critical for service mapping and impact analysis.

Key Insights:

- These tables form the foundation of the CMDB and enable organizations to manage their IT infrastructure efficiently.
- Understanding these tables helps IT teams maintain the integrity of the CMDB and support various processes like incident, problem, and change management.

Dependency View

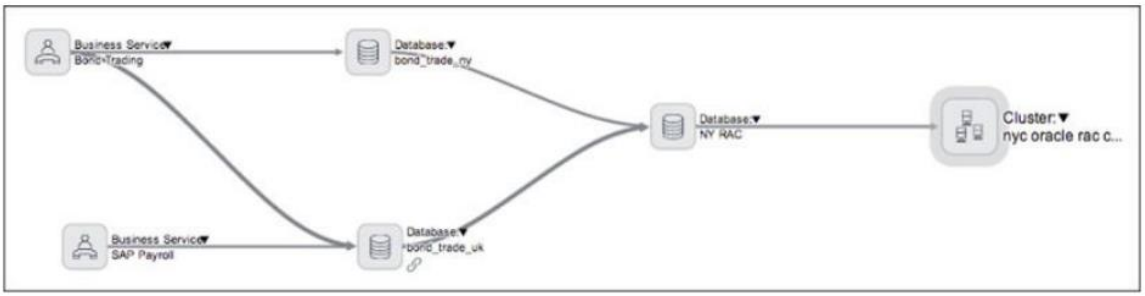
The **Dependency View** in ServiceNow is a visualization tool that helps in understanding the relationships and dependencies between various Configuration Items (CIs). This tool is crucial for impact analysis, troubleshooting, and managing changes within the IT environment.



Key Features of Dependency View:

1. **Graphical Representation**:
    - The Dependency View presents a graphical map of how CIs are interconnected. For instance, it can show how a particular server is related to the applications it hosts and the network it connects to.
2. **Service Impact Analysis**:
    - By visualizing dependencies, teams can assess how changes or incidents in one CI might impact other related CIs and services. This helps in planning changes with minimal disruption to services.
3. **Filter and Zoom**:
    - Users can filter the view to focus on specific types of CIs or relationships, and zoom in or out to see more or fewer details, respectively.

4. **Interactive Elements**:
    - The view allows for interactive exploration of the relationships between CIs. Clicking on a CI might provide more detailed information about that item or its relationships.



Example:

In a Dependency View, if a network router is affected, you can trace the impact on connected servers, applications running on those servers, and services dependent on those applications.

Key Insights:

- The Dependency View enhances visibility into the IT environment, supporting better decision-making and quicker issue resolution.
- It's particularly useful for IT service management processes like incident management, change management, and problem management.

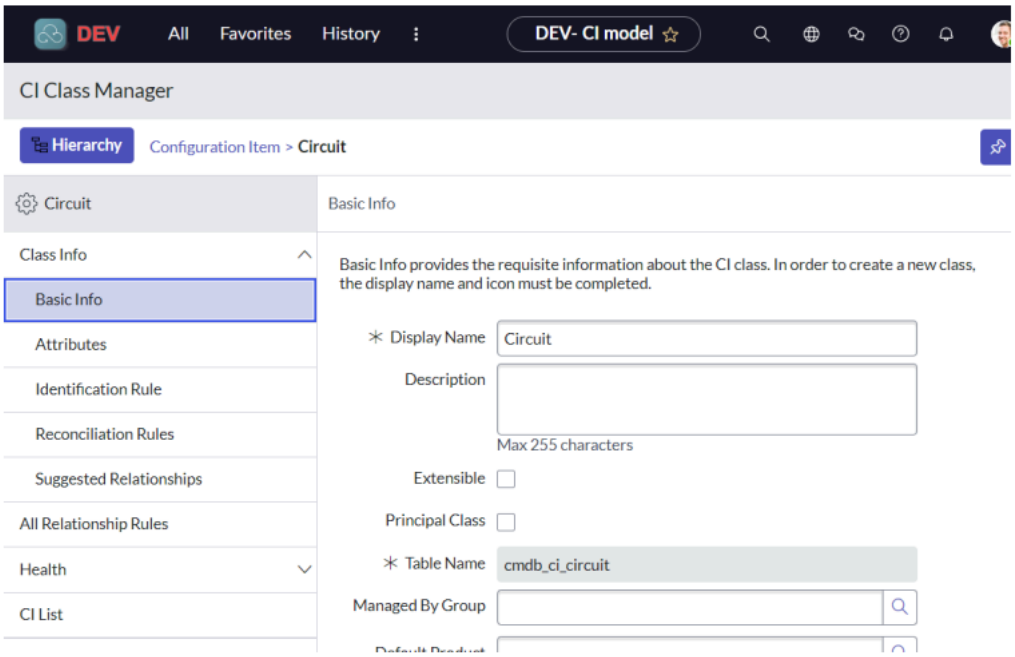
CI Class Manager

The **CI Class Manager** is a tool within ServiceNow that allows administrators to manage the Configuration Item (CI) classes within the CMDB. It plays a critical role in defining and maintaining the structure of CIs and their attributes.

Key Functions:

1. **Manage CI Classes**:
    - Administrators can create, modify, or delete CI classes based on the organization's needs. Each CI class represents a specific type of configuration item (e.g., servers, applications).
2. **Define Attributes**:
    - For each CI class, administrators can define and manage attributes or fields that are relevant for that class. For example, a server CI class might include attributes like CPU, RAM, and OS version.

3. **Set Relationships**:
    - CI Class Manager allows the setup of relationships between different CI classes, defining how different types of CIs interact with one another.
4. **Inheritance and Extension**:
    - CI classes can inherit fields from parent classes and extend functionality. For example, a **Database Server** class might inherit fields from the **Server** class and add additional fields specific to databases.



Example:

Creating a new class for **Network Devices** in CI Class Manager involves defining relevant attributes like IP address, device type, and location, and setting relationships with other CIs like routers and switches.

Key Insights:

- Proper management of CI classes ensures that the CMDB is structured in a way that reflects the organization's IT infrastructure accurately.
- It supports effective data collection, reporting, and impact analysis.

Base Info

Base Info in ServiceNow refers to the core information about Configuration Items (CIs) and other records stored in the CMDB. It includes fundamental details that define and describe the CIs.

Components of Base Info:

1. **Identification Information**:
    - Includes unique identifiers for each CI, such as **CI Name**, **Serial Number**, or **Asset Tag**.
2. **Classification**:
    - Classifies the CI into a specific category or class (e.g., Server, Application). This helps in organizing CIs and applying appropriate attributes and relationships.
3. **Attributes**:
    - Basic details such as **Manufacturer**, **Model**, **Location**, **Owner**, and **Status**. These attributes provide essential information about the CI's configuration and status.
4. **Lifecycle Information**:
    - Includes data about the CI's lifecycle stages, such as **Installation Date**, **Warranty Expiry Date**, and **Decommission Date**.

Example:

For a **Server CI**, base info might include attributes like **Server Name**, **IP Address**, **Operating System**, **Installed Applications**, and **Location**.

Key Insights:

- Base Info is essential for managing and maintaining CIs effectively, providing a foundation for other IT service management processes.
- Accurate base information helps in tracking CIs throughout their lifecycle and supports effective asset management and incident resolution.

Attributes

Attributes are specific pieces of information associated with Configuration Items (CIs) or other records in ServiceNow. They provide detailed characteristics and data that define each CI.

Types of Attributes:

1. **Basic Attributes**:
    - Fundamental details such as **Name**, **Type**, **Description**, and **Status**. These attributes are essential for identifying and categorizing CIs.

2. **Custom Attributes**:
    - Additional fields defined by users or administrators to capture specific information relevant to the organization's needs. For example, **Custom Field 1** or **Custom Field 2**.
3. **Computed Attributes**:
    - Attributes that are automatically calculated based on other data. For instance, **Uptime Percentage** might be computed based on recorded uptime and downtime.
4. **Reference Attributes**:
    - Fields that link to other records or tables within ServiceNow. For example, a **Manager** field in the **User** table might reference a record in the **Employee** table.

Example:

For a **Database CI**, attributes might include **Database Name**, **Version**, **Host Server**, **Storage Size**, and **Backup Frequency**.

Key Insights:

- Attributes provide the granularity of information needed for detailed reporting, analysis, and management of CIs.
- Properly defined attributes ensure that all relevant data is captured and available for IT service management processes.

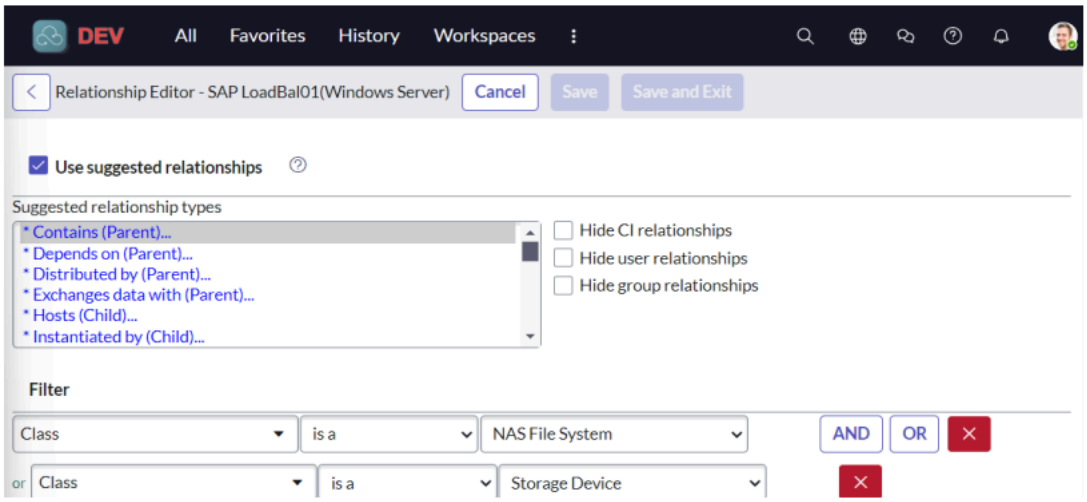
Suggested Relationships

Suggested Relationships in ServiceNow refer to the automatic recommendations for linking Configuration Items (CIs) based on predefined rules or patterns. These relationships help in visualizing how different CIs are interconnected.

Types of Suggested Relationships:

1. **Automatic Relationships**:
    - Based on CI types and their attributes, ServiceNow can automatically suggest relationships. For instance, a **Server** might automatically be linked to its **Storage Device** based on common attributes.
2. **Rules-Based Relationships**:
    - Relationships are suggested based on specific rules or criteria defined by the organization. For example, if two CIs share the same **Location**, a relationship might be suggested.

3. **Manual Review**:
    - Suggested relationships can be reviewed and adjusted manually by administrators to ensure accuracy. This is particularly useful for complex or custom relationship types.



Example:

If a **Web Server** CI is added, ServiceNow might suggest relationships to associated **Application** CIs and **Network Device** CIs based on predefined rules.

Key Insights:

- Suggested relationships enhance the accuracy and completeness of the CMDB by ensuring that all relevant connections between CIs are identified.
- They support better impact analysis and decision-making by providing a clearer view of the IT environment.

Update Sets

Update Sets in ServiceNow are used to package and move customizations and configurations between different instances of ServiceNow, such as from a development instance to a production instance.

Key Features of Update Sets:

1. **Customization Packaging**:
    - Update Sets bundle together changes made to forms, tables, business rules, and other customizations. This allows for consistent deployment of customizations across different instances.
2. **Tracking Changes**:

- Update Sets track all changes made during development, including additions, modifications, and deletions. This helps in maintaining a clear record of what has been changed and why.

3. **Deployment**:
    - Once an Update Set is completed and tested in a development instance, it can be moved to a test or production instance. This ensures that customizations are applied in a controlled manner.
4. **Merge and Conflict Resolution**:
    - When multiple Update Sets are applied, conflicts can occur if changes overlap. ServiceNow provides tools for merging Update Sets and resolving conflicts.

Example:

An Update Set might include customizations such as a new business rule for handling incidents and changes to the incident form layout. This Update Set would be moved from a development instance to a production instance to apply these customizations.

Key Insights:

- Update Sets streamline the process of migrating customizations between instances, ensuring that changes are deployed consistently and reliably.
- They are essential for maintaining version control and managing customizations in a multi-instance environment.

Business Rules

Business Rules in ServiceNow are server-side scripts that execute automatically in response to database operations (such as insert, update, or delete). They are used to enforce business logic and automate tasks.

Key Components of Business Rules:

1. **Trigger Conditions**:
    - Define when the Business Rule should run. Triggers can be based on operations such as **Insert**, **Update**, **Delete**, or **Query**.
2. **Actions**:
    - Actions are the tasks performed by the Business Rule, such as updating a record, sending notifications, or calling a script.
3. **Script**:
    - Contains the logic that defines what the Business Rule does. For example, a script might calculate a field value based on other fields or perform complex data manipulations.
4. **Execution Order**:
    - Business Rules are executed in a specific order (pre-update, post-update, etc.), allowing for control over when different actions occur in relation to the database operations.

Example:

A Business Rule might be set to automatically assign a priority level to an incident based on its impact and urgency. This rule would trigger when an incident is created or updated, and it would update the **Priority** field accordingly.

Key Insights:

- Business Rules enable automation and ensure that data is processed consistently according to business requirements.
- They are crucial for implementing business logic and workflows within the ServiceNow platform.

NAME : MADAMANCHI GEETHIKA

COLLEGE ID NO : 2100031164

Cognizant Reg Email : 2100031164cser@gmail.com