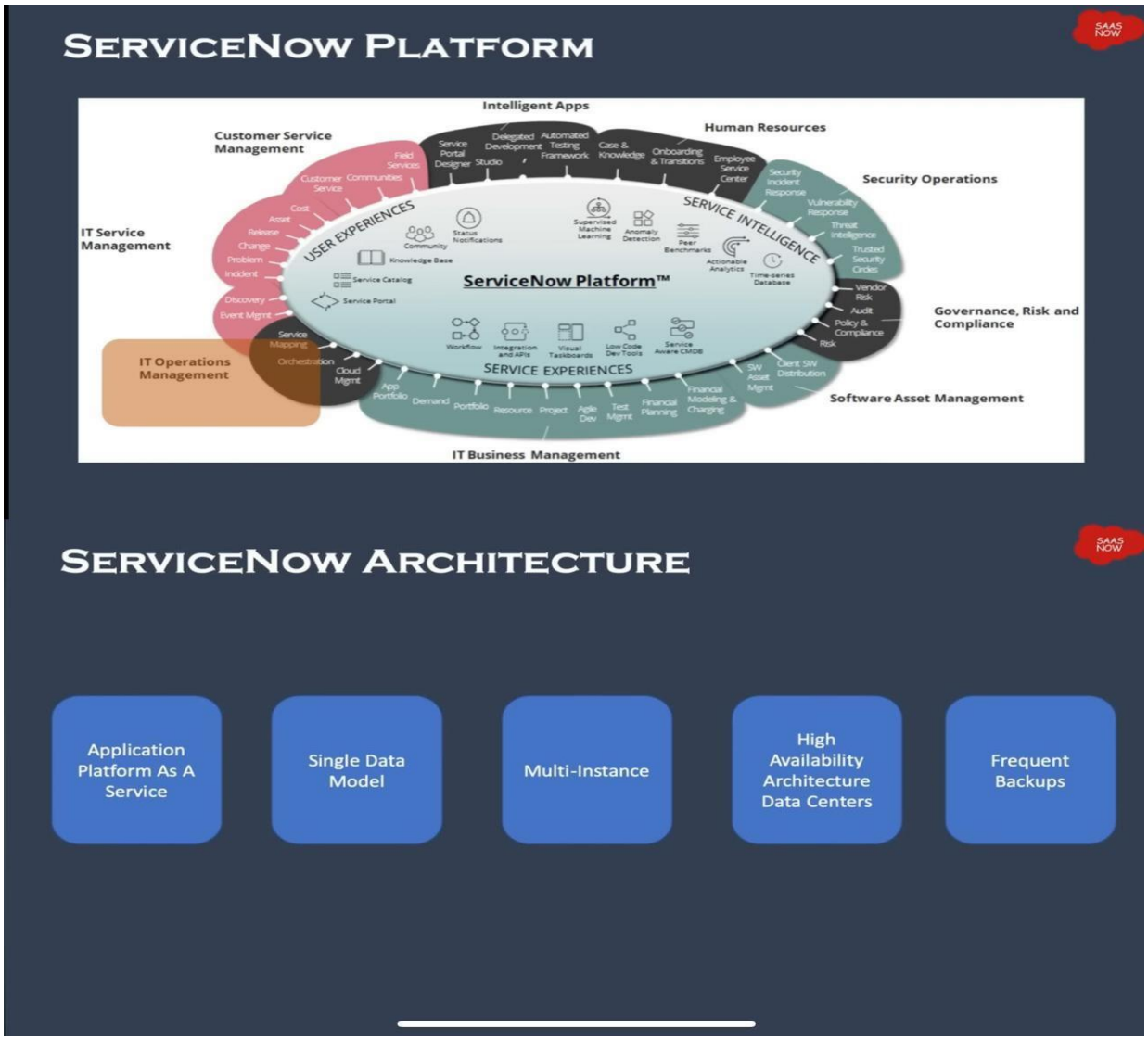
WEEK-2

ServiceNow Platform Overview

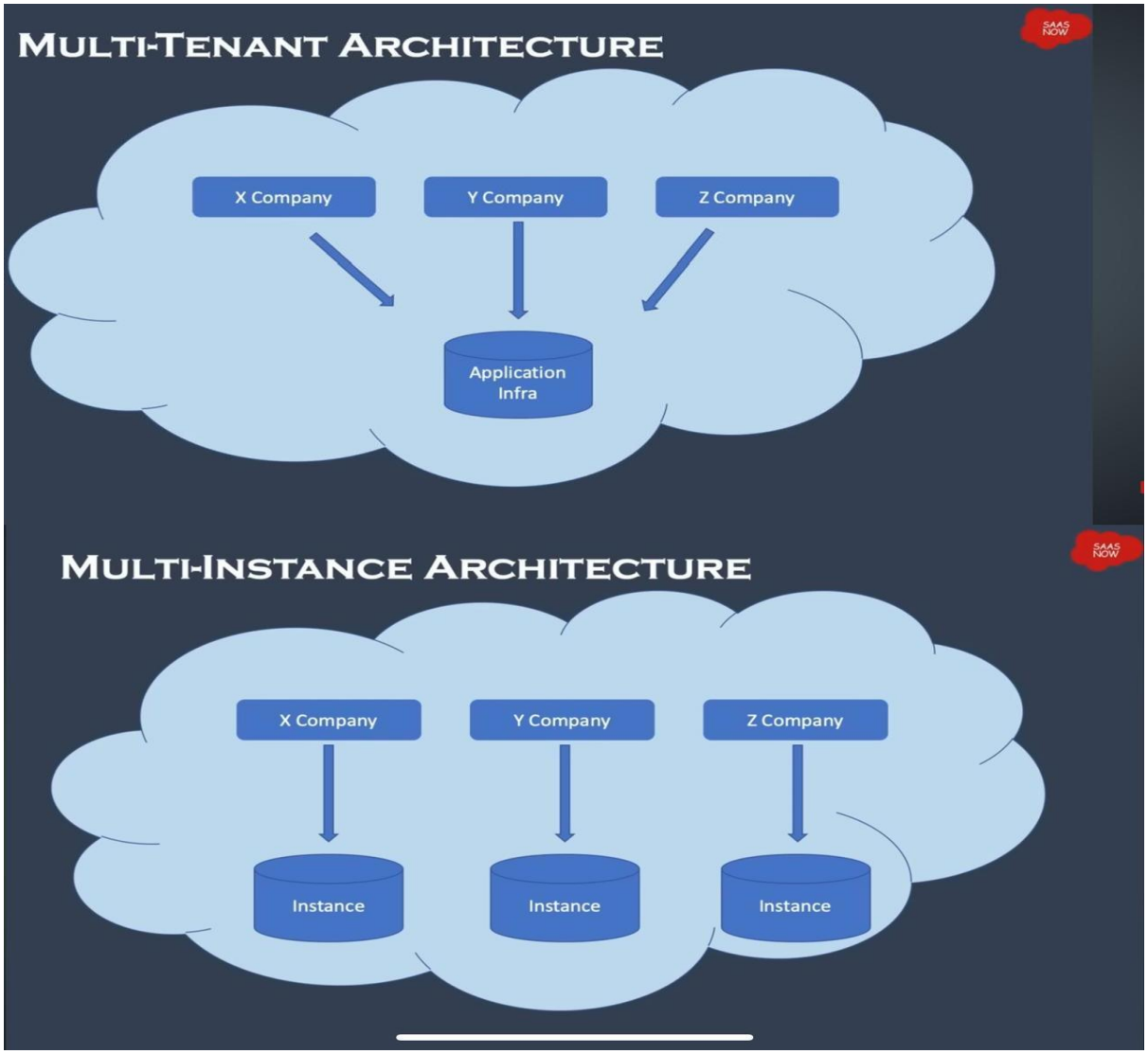
The ServiceNow Platform is an **Application Platform-as-a-Service.**



- ServiceNow utilizes an **advanced, multi-instance, single-tenant** architecture as the default offering for customers, meaning an instance features an individually isolated database containing data, applications, and customizations.

- ServiceNow provides services to its users from a configurable **web-based user interface, built on top of a flexible database schema.**

- The Platform and the applications that run on it use a **single system of record** to consolidate an organization's business processes.
- The Platform integrates with other enterprise systems and supports a wide variety of **plug-and-play applications**.
- ServiceNow Provides a platform upon which you can build custom applications.
- All ServiceNow Data Centers are paired with another datacenter to provide redundancy. **Redundancy is built into every Layer** including devices and network resources
- **Backups & Security** ,Servicenow provides **4 weekly** full data backups and **6 days** ofdaily differential backups. The entire platform is secured using third party security organization



Authentication in ServiceNow

1. local database authentication
2. External single sign-on(SS0)
3. Multi factor authentication
4. Digest Token
5. OAuth 2.0

Types of Instances

There are mainly **two** types of instances

1.Production

2.Non Production - it has development , testing, quality assurance

In ServiceNow, a **PDI** stands for **Personal Developer Instance**. It is a free, individual instance of the ServiceNow platform provided to developers for learning, experimentation, and building applications.

Each ServiceNow instance has a **unique URL** that uses a format similar to

https://<instance name>.service-now.com

Roles In ServiceNow

User- A user in ServiceNow is any individual who needs access to the platform. Each user is uniquely identified by a user ID.

They are stores in **sys_user** table

Group- A group is a collection of users who share similar responsibilities or perform similar tasks.Groups make it easier to manage permissions and assign tasks.
They arestored in **sys_user_group** table

- **Role**- A role defines a set of permissions that control what a user can access and what actions theycan perform within ServiceNow. Roles can be assigned to both users and groups.

They are stores in **sys_user_role** table

SERVICENOW USER INTERFACE OVERVIEW

3 Main Screen Elements

1.**Banner Frame**-**Logo**;**UserMenu**(profile,Impersonate User(Access to admin or impersonate user),Elevate Roles,Logout);Tools(**Globalsearch**-record matching keyword,**conectchat**-realtime messaging,**help**-contextual help,access to user guide and documentation search);**System Settings**(Customize the UI to their preferences)General,**Theme,Accessibility,List,Forms,Notifications,Developer settings**.

2.**Application Navigator-**Navigation Filter,All Applications,History(30 items),Favorites.

3.**Content Frame**

SERVICENOW BRANDING OVERVIEW

What is Branding in ServiceNow?

Applying your distinct corporate identity across the Now Platform UI to create a shared identity,build trust and speed adoption

Guided Setup

A System Administrator step-by-step instructions to configure various Applications and Modules within your instance to suit the needs of the users.

1.ApplicationNavigtaor>Guided Setup>**ITSM Guided Setup**

Company-System Configuration,Welcomepage ,Connectivity,Foundation Data,CMDB,Incident Management,Major Incident Management,Problem Management,Change Management,Service Catalog,Knowledge Management,Continual Improvement Management,Project Communication,Go Live.

2.ApplicationNavigtaor>Guided Setup>**ITOM Guided Setup**

MID Server,Discovery,Event Mnagement,Operational Intelligence,Cloud Provisioning and Governance

Service Portal And UI Builder-tools used to brand interface

Service Portal-Widget based tool

UI Builder-Build Functional page (buttons and DV) and layouts

LISTS AND FILTERS

Lists

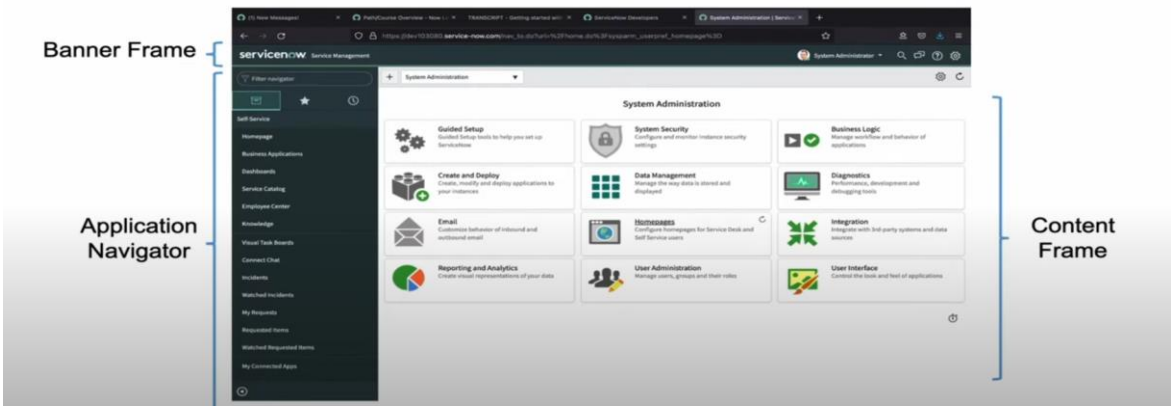
Is a user interface page displayed in the content frame that's designed specially to present lists of records from database tables it includes tools that make it esy to sort search filter and analyze list data quickly and simply it also provides the ability to select a single item from the list to display in more detail via form view there are many to access lists in servicenow –the application navigator

Incident>All—opens the list interface for the incident table

Tablename.list—open the list interface for that table

Task.list,incident.list,sys_user.list

To use dot list command user must know the name of the table if not known the names of table sys_db_object.list—will open the tables table,sys_db_object is the name of a table in the servicenow database that stores a record for each table in the servicenow database.



Tables and columns in ServiceNow

Sys_db_object - table used to store information about all the tables in servicenow

Sys_dictionary - table used to store information about all the fields of all the tables in

servicenow

Sys_documentation - tables used to store all the field labels in servicenow

The **System Dictionary** in ServiceNow is a core component that defines and manages the structure of the database tables and their associated fields. It acts as a central repository where information about the database schema is stored, including the definitions of tables, fields, datatypes, and relationships between tables.

The System Dictionary contains the definition for every field from all tables in the ServiceNow instance.

All > System Definition > Dictionary to access the system dictionary to modify table and field attributes.

All > System Definitions > Tables or All > System Definitions > Tables & Columns to review or create new tables.

Notifications

A notification is a tool for alerting users when events that concern them have occurred.

They can be triggered by events in the platform and require no scripting knowledge. Use notifications to

notify users about activities in ServiceNow (i.e., updates to incidents or change requests).

The following notification methods are used in ServiceNow:

- Email
- SMS
- Meeting Invitation

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.

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

The default recipients for message is 100, if we want to send it to 1000 people then it will send the msg 10 time

If you want to change the recipient limit, set the system property**glide.email.smtp.max_recipients.**

Email Layouts

Emails are created to specify the HTML content you want to appear in the body of one or more email templates. By default, the system includes several sample layouts administrators can use to create their own layouts. 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

Knowledge Management

KM involves creation, sharing, viewing or knowledge articles that are used to provide information to self users and process users for their day to day works.

Knowledge Base contains Categories , Categories Contain Knowledge Articles
To View Knowledge Articles

ALL - self-service - knowledge

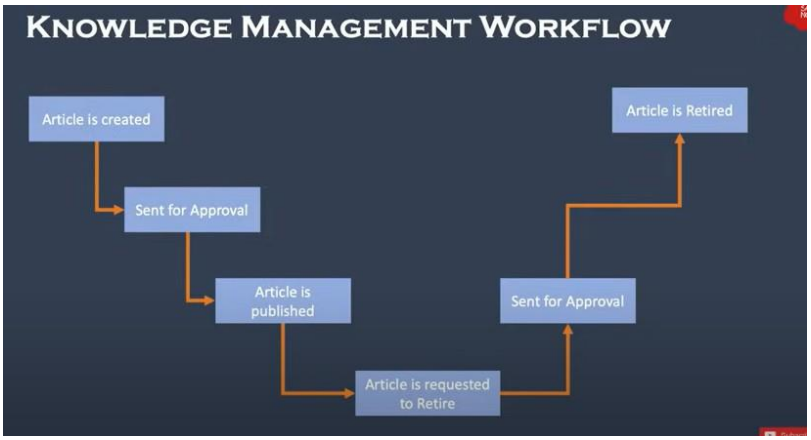
ALL - Knowledge - Homepage - opens workspace containing all the Knowledge basesALL -

Knowledge - All - open table containing all the Knowledge Articles

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

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 andCategory. 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.

Administrators can create multiple Knowledge Bases and assign them to individual managers responsible for controlling the behavior and organizational schema of each Knowledge Base

Knowledge management **Guided setup** is used to develop Knowledge Base for theorganization

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 - approveOr

As system administrator you can open Knowledge article records - scroll to related lists Approvals - Approve .

User Criteria

User Criteria defines conditions that are evaluated against users to determine which users can create, read, write, and retire knowledge articles.

- You can apply several user criteria records to knowledge content.
- User Criteria is applied at the **Knowledge Base level**.
- If a Knowledge Base has no user criteria selected, articles within that Knowledge Base are available to all users.

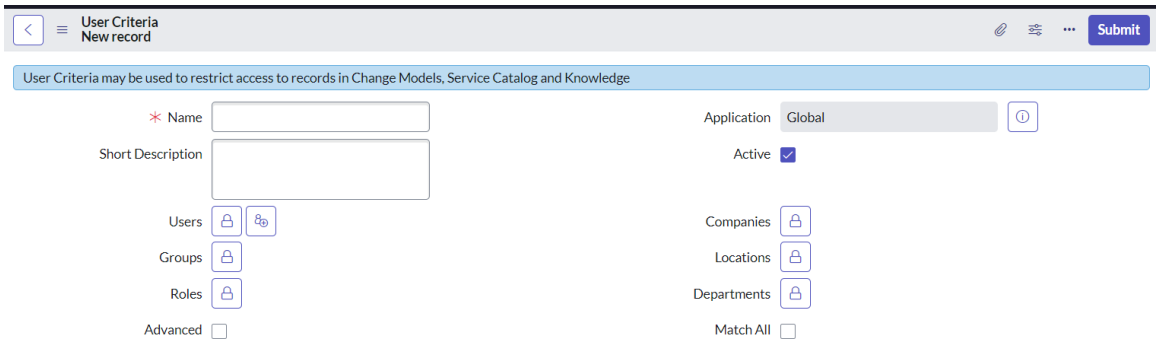
To control access to logged in users only, administrators should leverage the glide**.knowman.block_access_with_no_user_criteria** property.

User Criteria outcomes include:

- canRead: users who can read all Knowledge Base articles
- cantRead: users who cannot read, create, or modify articles in the Knowledge Base
- canContribute: users who can read, create, and modify articles in the Knowledge Base
- cantContribute: users who cannot create or modify articles in the Knowledge Base

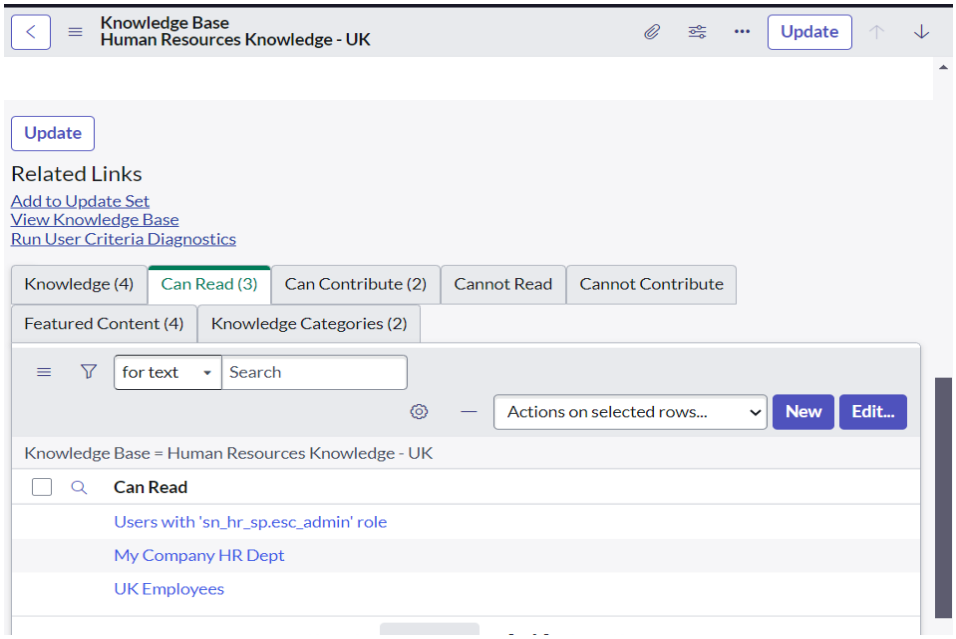
To create a User criteria

All - knowledge - User criteria - New



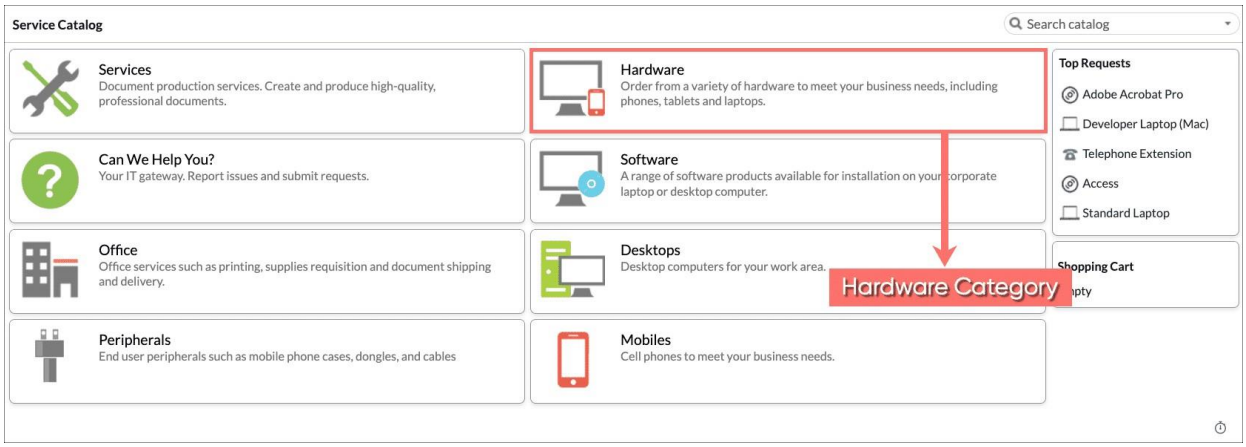
To implement user criteria, navigate to

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

One stop shopping for ordering, requesting required products , Services. The ServiceCatalog application in the Platform allows users to view, request, and shop around for services and products.



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

All > Service Catalog > Catalog Definitions > Maintain Items

Variables and Variable Sets

The Service Catalog variables help define the structure of a catalog item form that is displayedto the customer. For example, you can define variables as Hardware Type, Color, or Price, etc.

Functionally, a Variable Set is just a container, so it has only two fields: Name and Description. To

create a new variable set.

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

Common Variable Types

- Multiple Choice: Creates radio buttons for user-defined question choices
- Select Box: Creates a choice list of user-defined question choices
- Single Line Text: Creates a single-line text input field
- Reference: Specifies a record in another table, similar to a reference field
- Check box: Creates a check box which may be selected or cleared

Record Producer

A Record Producer focuses on a specific process or task and can be used anywhere in the ServiceNow platform.

A record producer is a specific type of catalog item that allows end users to create task-based records, such as incident records, from the service catalog.

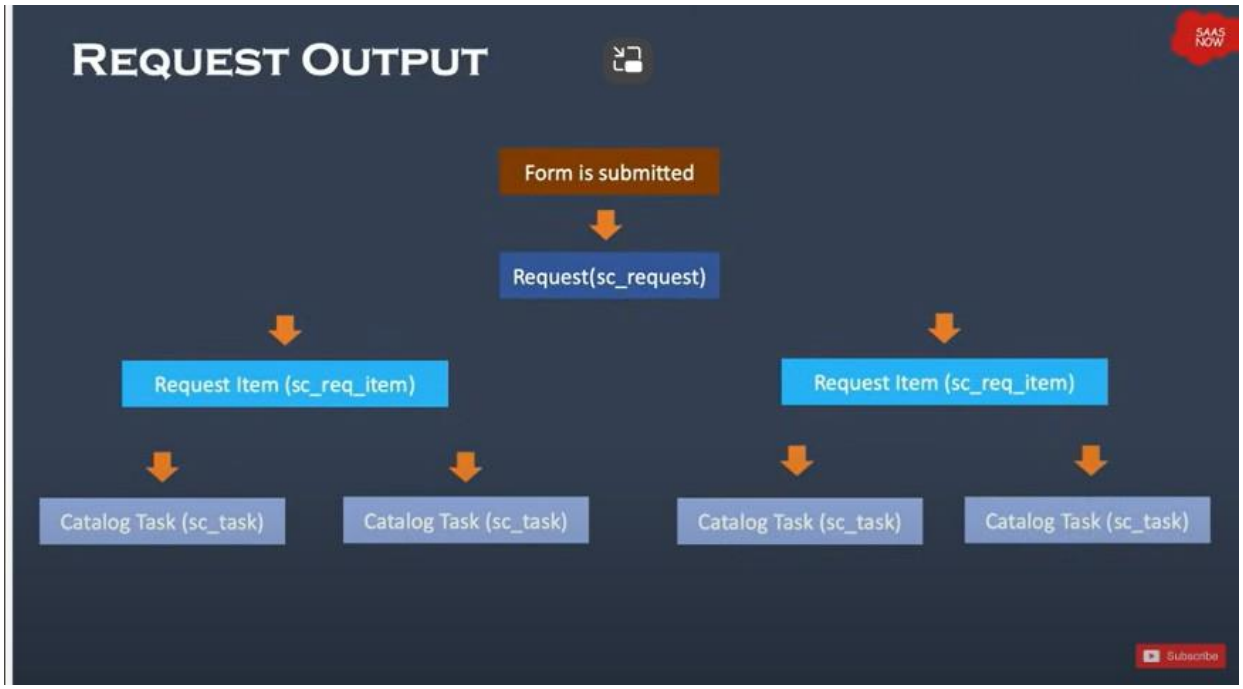
Record Producers appear as simplified forms, allowing users to provide information that is translated into task-based records being added or modified in the database.

Order Guide

Order Guides provide the ability to order multiple, related items as one request. Remember that variables are presented by the Order field number. Use an Order Guide to assist users in determining what items they need.

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).

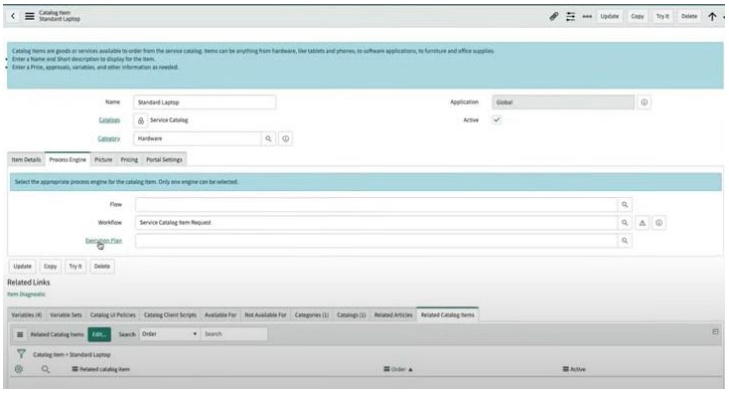
SCTASK# Catalog Task [sc_task] table: Records on this table begin with SCTASK and are the assigned tasks needed to complete the delivery of each Request item from start to finish. Some of the more important fields are the Assignment group, the Due date, Work start, and Work enddates.

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:

- Waiting for approval (In Progress)
- Approved
- Pending (has not started)
- Fulfillment (In Progress)
- Deployment/Delivery
- 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.

Table management

Everything in servicenow is built on a relational database provided by servicenow platform Records are identified by a **32-character**, globally unique ID, called a **sys_id.**

Administrators can use these tools for viewing and modifying the database structure:

Tables module: Provides a list of all tables in the database.

Tables & Columns module: Provides a list of all existing tables, with columns, column attributes, and indexes.

Schema map: Provides a graphical representation of the relationships between tables.

Data dictionary tables: Contains additional information that defines database elements.

Field Labels and field names are different

Types of tables

Core Tables:

- Description: Core tables are the fundamental tables provided by ServiceNow out of the box. These tables are integral to the platform's functionality A core table is something that comes with the Service now base system.
- Examples:
  - Task (task): A core table used as a parent for many other tables like Incident, Problem, Change Request, etc.
  - User (sys_user): Stores user records,CMDB (cmdb_ci): Core table for Configuration Items (CIs).

Custom Tables:

- Description: Custom tables are user-created tables. When creating a new custom table, the table name is automatically populated based on the table label and a prefix. If the table is being created in a scoped application, the name is prefixed with a namespace identifier: **"x_", indicating that it is a part of an application. Otherwise, custom tables in the global application feature "u_" as their prefix, and then the table name.**
- Examples:
  - A table to track internal projects (u_project), A table for storing customer feedback (u_feedback).

Extended Tables

Description: Extended tables are tables that inherit fields and behaviors from a parent table. This is part of the ServiceNow table inheritance model.

- Examples:
  - Incident (incident): Extends the Task table,
  - Problem (problem): Also extends the Task table.

Base Tables:

- Description: A base table is a table that is not extended from any other table. It is at the top of its table hierarchy. Many core tables in ServiceNow are base tables.
- Examples:
  - Task (task): A base table used to manage tasks.
  - CMDB (cmdb_ci): Base table for Configuration Items.

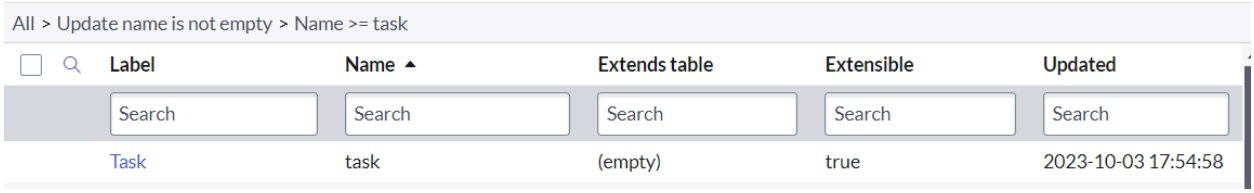
System Tables:

- Description: System tables store data that ServiceNow itself uses to manage its operations, such as user records, roles, settings, and more. These tables are often hidden from the standard user interface.
- Examples:
  - sys_user: Stores user records.sys_db_object: Stores metadata about tables inthe instance. sys_dictionary: Stores the definitions for fields in tables.

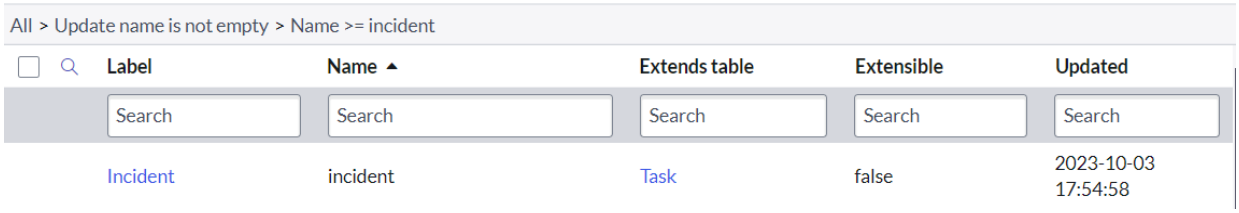
The two properties of tables

Extends table - determines from which table the current tables is extended Extensible - is true

/ false field that determine if any other table are extended from it



For Extended table - Extends table is the name of the table it is extended from and extensible is false if any other table is not extended from it.



Schema Map:

- **Schema Map**: The Schema Map is a visual tool in ServiceNow that shows the relationships between tables, including which tables are parents, which are children, and how they relate to each other.

All - System Definition - Tables & columns - Schema Map - will open in new window

It shows the complete outline of a table and its relationships . It will show what are the referenced tables , referencing tables, extended tables, extending tables.

Extending tables - tables that are extended from the current table, tables with blue bars. (childof current table)

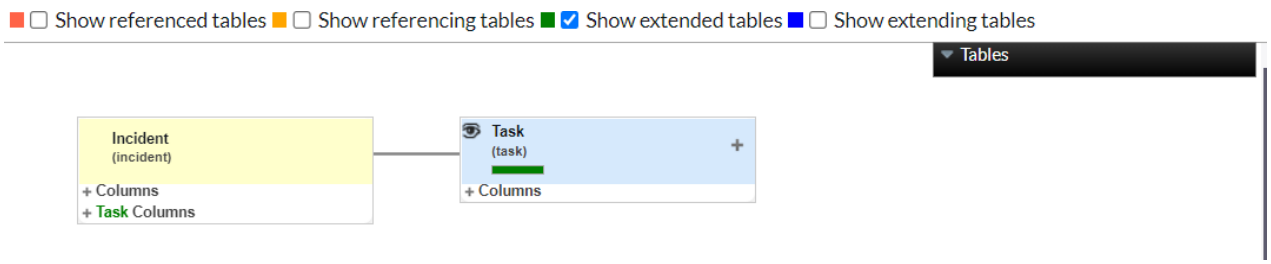
Extended tables - table from which current table is extended from, green color (parent tables)

Referenced tables - tables that current table refer to for records

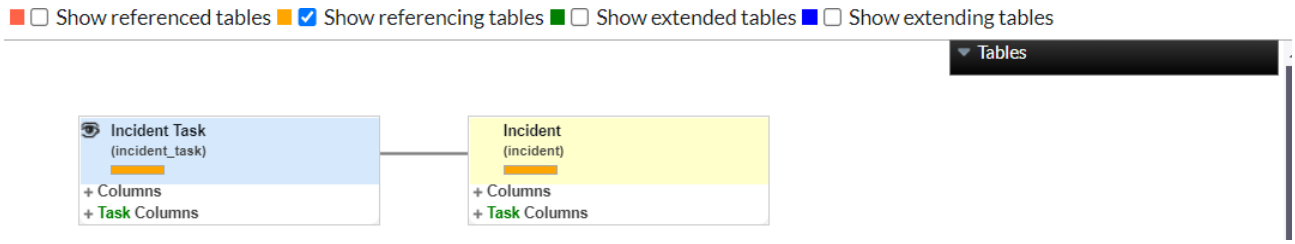
Referencing tables - tables that have fields that refer to records in current table

Schema map of Incident Table

Extended table is task, as it is extended from it



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



Types of table relationships

One to many

Many to Many

Extensions

Database Views

One-to-Many - A single record in the parent table is related to multiple records in the child table(e.g., Users and Incidents).

There are three one-to-many relationship fields:

1. Reference Fields - Allows a user to select a record on a table defined by the reference field. Example: Caller field on the Incident table allows a user to select any record on User table

2. Glide List -Allows a user to select multiple records on a table defined by the glide list. Example: The Watchlist field on the Incident table allows the user to select any record or records on the User table.

3. Document ID Fields - Allows a user to select a record on any table in the instance. Example: Document field on the Translated Text table.

Many-to-Many - Bi Directional Relationship - Multiple records in one table relate to multiplerecords in another table, managed by a joining table (e.g., Users and Groups).
Sys_m2m and sys_collections tables information about m2m tables.

Database views - They are used to combine two tables for reporting an analysis based on a common field . Having a common field is essential for generating database views in ServiceNow. It is the same as joins in SQL . The data in the virtual table created by a databaseview is read-only.

Create Database Views by navigating to System Definition > Database Views.

Extensions: A child table inherits fields from a parent table (e.g., Task and Incident). A table that extends (is an extension of) another table is a child class. The table from which it extends is theparent class.

Access Control List -

it determines how the servicenow user is going to interact with the Data .It is stores in tables **sys_security_acl**

There are three security modules typically used by the System Administrator:

- **All > System Properties > Security**
- **All > System Security > Access Control (ACL)**
- **All > System Security > High Security Setting**

Access Control

It is a security imposed on tables to restrict users to interact or modify with the data of the table,It restricts the use of CRUD operations. it is applied on two levels

Row level , column level

Other than CRUD it is also restricts service now specific operations to be performed

1. Personalize choice
2. Edit_ci_relations - user cannot define relationship between configuration tables
3. Report on - user cannot create reports
4. Execute - run scripts or UI
5. Save_ templates - controls data when template is saved

ACL - Access Control List - It contains all the Access Controls of that particular Instance

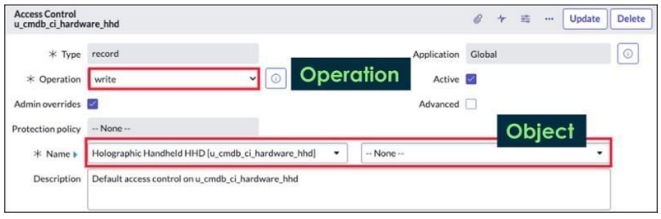
To view ACL

ALL - System Security - ACL

ALL - Table_Name.CONFIG - Access Controls of table

Each Access Rules Specifies three components

1. Operation - valis servicenow function
2. Object - table, record, field
3. Permissions



Access Control : Rule Types

1. table.–None–
2. table.field
3. table.*

1

2

3

1. No specific field selected - this rule applies to the whole table including all its records

2. **This rule applies to only one field on a record and in this case, the Caller field on an incident record**

3. Wildcard – this rule applies to every field on a record without a table.field rule. It reduces the acls to be written

When we take the house as an example, the table.none- is the whole house, table.field is aparticular room and table.* is all the rooms except the table.field room

Table.none - this used to grant access at **record level**, ie it grants access to view and edit records

table.* - this is used to grant access at the **field level,** if permission is granted users will access the fields but not the records

Creating a ACCESS CONTROL

All - System Security - Access Control List - New - Select the table and field - Save -Scroll down
Add Role - submit

The Acl will be reflected in the table related lists section

ALWAYS ADD ROLES TO ACL - ROLES TO GROUPS - USERS TO GROUPS

This way users in the group have access to that ACL

Elevate Role

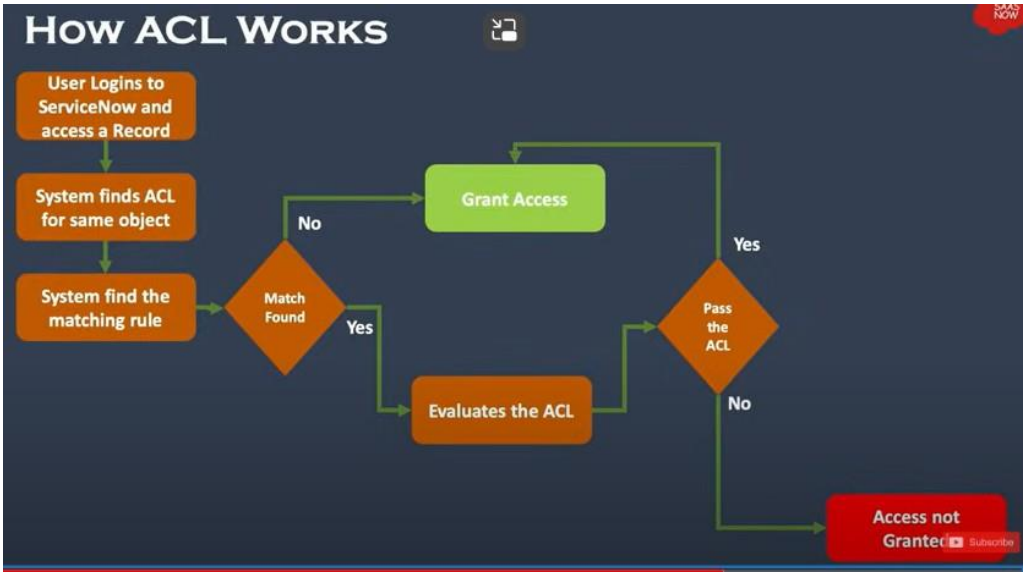
The base system admin can elevate to a privileged role to access features of High Security Settings. To elevate roles:

1. Open the user menu
2. Select Elevate role
3. Select an elevated role and click Update

Evaluating ACL

Table ACL Rules: These are checked first and determine whether the user can access the entire record in the table. The system evaluates these rules from the most specific to the most general.

Field ACL Rules: These are checked after the table ACLs and determine whether the user can access specific fields within the record. The system also evaluates these rules from the most specific to the most general.



- If a user fails a table access control rule, the user is denied access to all fields in the table, even if the user would pass a field ACL rule.

- If a user passes a table ACL rule, but fails a field ACL rule, the user cannot access the field described by the field ACL rule.

Import Sets In Servicenow

They are used to load data into tables in servicenow from different sources

Import Sets provide a mechanism to pull data into ServiceNow. Import Sets store data in Import Set tables. Any user logged in with **the admin or import_admin role can manage all aspects of Import Sets.**

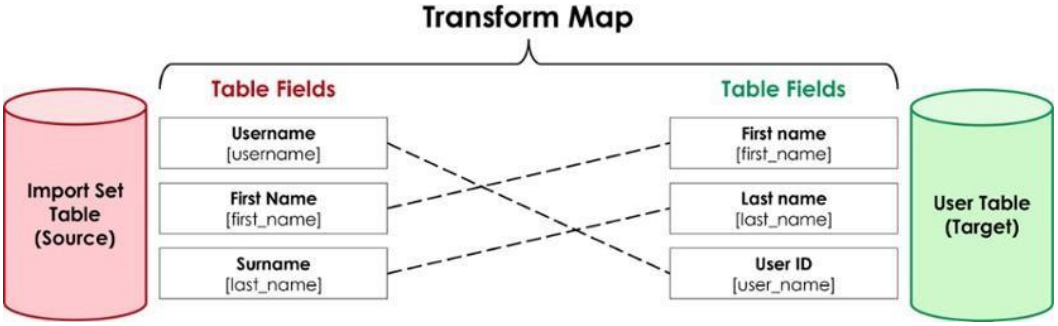
In service now we can not directly load data into tables , Thus we use the Import sets

There are 6 components to import data to table say incident in servicenow

The **Import Set Table acts** as a staging area for records imported from a data source.

Transform Maps provide a guide for moving data from Import Set (staging) tables to "Target" tables. Field mapping provides direct field-to-field data moves.

A transform map is a set of field maps that determine the relationships between fields in an import set and fields in an existing ServiceNow table, such as Incidents [incident] or Users [sys_user].



Coalesce Fields

Coalescing a field (or multiple fields) means the field will be used as a **unique key during imports.**

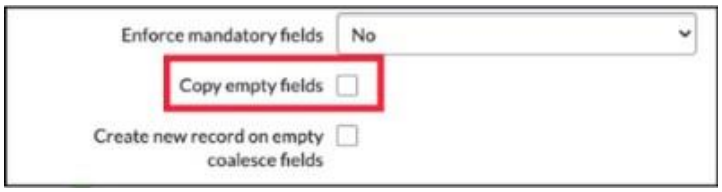
If a **match is found** using the coalesce field(s), the existing record will be updated with the information being imported

If a **match is not found** using the coalesce field(s), then a new record will be inserted into the database

There are three types of coalesce

1. Single
2. Multiple
3. Conditional - script is written to return sys_id

On the Transform Map form, locate the option for Copy empty fields.



There are two types of mapping done in Transform Map

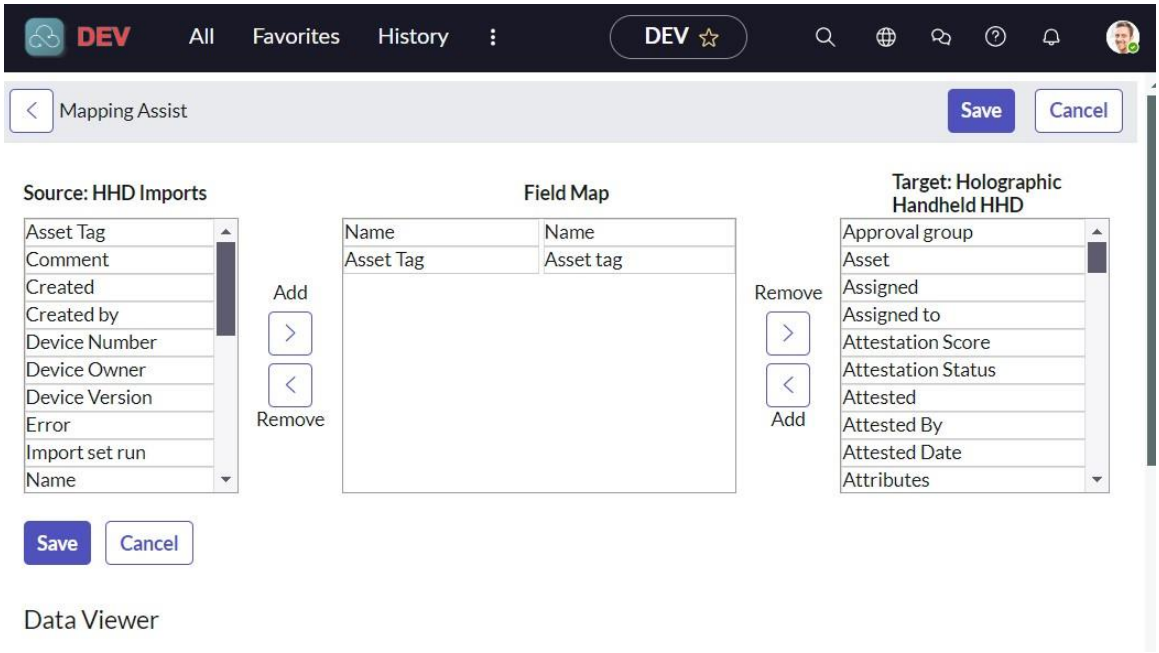
Automatic Mapping Utility: field names of the Import Set match the name of the fields on the**Target table** where the data will be transformed. In this case, simply click Auto Map Matching Fields in the related links in the Table Transform Maps form and confirm proper matching.

Mapping Assist Utility: The Mapping Assist utility provides a visually intuitive environment for specifying mapping between Import Set fields and Target table fields. With the Mapping Assistutility, it is possible to map a single source field (field on an Import Set table) to multiple destination fields (fields on a Target table). If there are any discrepancies in terms of how fields were automatically matched, these can easily be corrected using the Mapping Assist utility.

When all fields are matched properly**, click Transform in the related links** to begintransforming data onto the destination table.

Process to import data into servicenow table from excel

All - System import sets - load data - creating import set table - choosing the file - loading datainto import set table - open the import set table - go to related links - transform map -
Assist mapping - select the Servicenow table - map the fields - save - Transform



The following steps (process) can be completed by any user with the role import_admin or import_set_loader and import_transformer.

you can also use import option in Column options menu for excel and import XML for XMLdata

CMDB (Configuration Management Database)

The **Configuration Management Database (CMDB)** in ServiceNow is a centralized repository that stores information about all the **Configuration Items (CIs)** within an organization's IT environment.

A **Configuration Item (CI)** is any component within an IT environment that needs to be managed to deliver an IT service. Each CI in the CMDB has a set of attributes and relationshipsthat describe its characteristics, status, and how it interacts with other CIs

The Configuration Management Database is a series of tables and fields that contain all of the Configuration Items (CIs) , their attributes and relationships. Access to the CMDB tables and underlying data requires certain permissions, such as the following roles:

Three key CMDB tables are

1. **Base Configuration Item [cmdb]** - it is the parent table of all CI both **IT and Not IT**
2. **Configuration Item [cmdb_ci] -** which contains CI Data of **IT r**elated CIs , Its is extended from cmdb
3. **CI Relationship [cmdb_rel_ci] -** which contains CI relationship data.

CMDB is used to effectively manage the root cause of the problems caused by theinfrastructure failure of the organization.

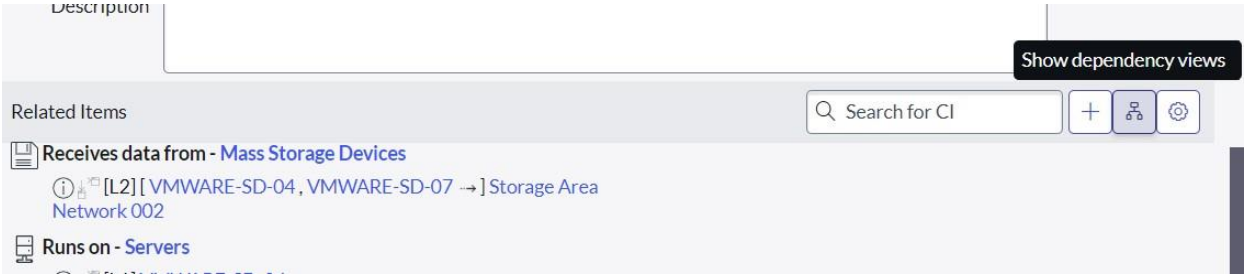
An accurate, up-to-date CMDB helps IT teams to:

- Locate failed changes and associated **Incidents**
- Facilitate impact analysis of proposed **changes**
- Assess problem trends pertaining to **specific CIs**
- Efficiently manage incidents affecting CIs and service delivery

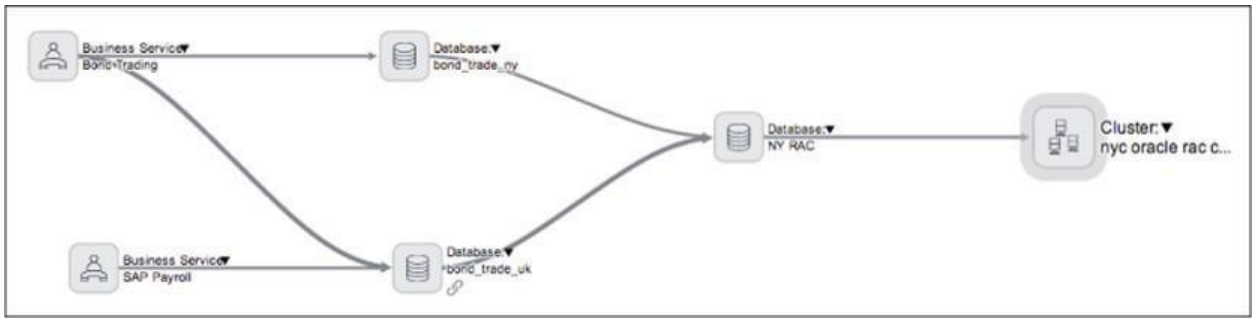
Dependency View

Dependency Views provide an interactive graphical interface to visualize relationships between configuration items.

ALL - Configuration - Open a CI - Form View scroll to Related Items



In Dependency view the **root CI or root node** is represented as a dark pulsating icon at the center, Dependency view shows both upstream and downstream relationships, by default theDependency view shows **3 levels**



CI Class Manager

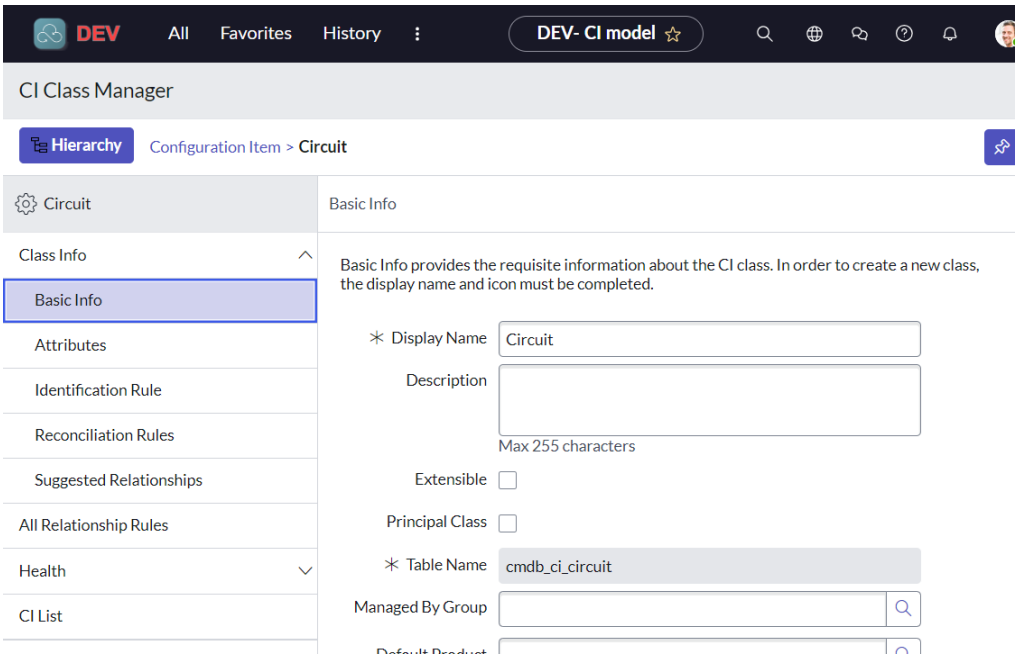
The CI Class Manager in ServiceNow is a feature that allows administrators to manage the Configuration Item (CI) classes in the CMDB.

The CI Class Manager displays the entire **CMDB class hierarchy in a tree-view format**.

A CI Class represents a type of Configuration Item, essentially a table that collects specific data

A CI Class in ServiceNow is a category or a type of Configuration Item (CI) that shares common **attributes and properties in the Configuration Management Database (**CMDB). Each CI class represents a specific kind of asset, component, or entity that an organization wants to track andmanage within its IT infrastructure.

You can also select a specific class to view. For each class, you can directly access CMDB Health settings, identification and reconciliation rules, CI list, Relationship rules.



Basic Info: Displays details for the selected class, such as the display and table name, description, and class icon.

Role required: itil for reading, and itil_admin and personalize_dictionary for writing.

Attributes: Displays table attributes (columns). You can edit those attributes and add new ones. It has

All, Derived and Added types

Role required: itil for reading and itil_admin and personaloze_dictionary for writing

Suggested Relationships

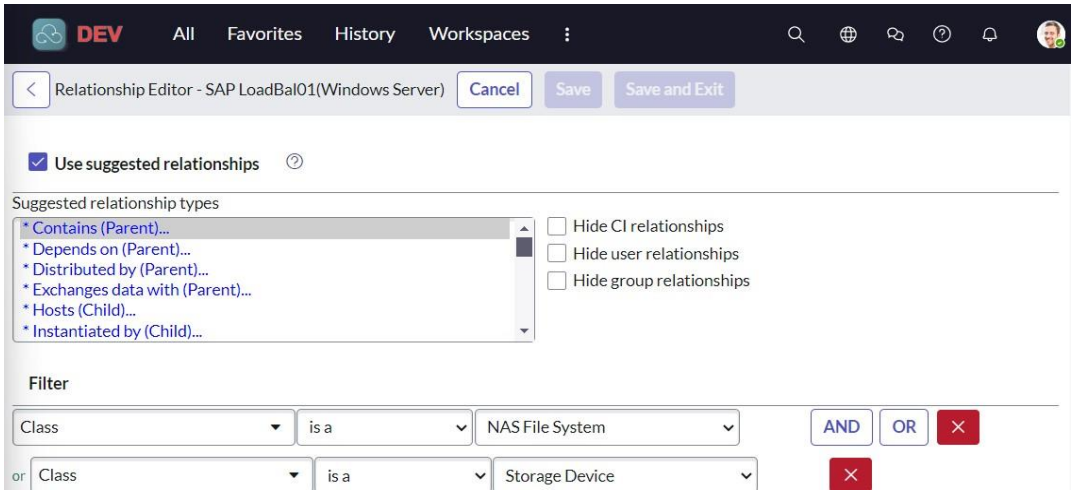
Use the CI Relationship Editor to create configuration item relationships.

CI Relationship Editor is accessed from the **Related Items toolbar** on a configuration item form.

The CI record where the editor was launched is designated as the base CI. Depending on the selected relationship type, the base CI can become the **parent CI or the child CI** in a new relationship.

You can create a new relationship rule in

All > Configuration > Relationships > Suggested Relationships.



Update Sets

An Update Set is a group of configuration changes that can be moved from one instance to another.Update Sets allow administrators to group a series of changes into a named set and then move them as a unit.

Every instance of ServiceNow has a default update set, however, admins should use named update sets for moving customizations between instances.

An update set is an XML file that contains:

- A set of record details that uniquely identify the update set.
- A list of configuration changes.
- A state that determines whether another instance can retrieve and apply configuration changes.

Update sets track changes to applications and system platform features.

Basically, an Update Set record is a “point in time” XML snapshot of process records. An Update Set works by writing changes from tracked tables to the Customer Update **[sys_update_xml]** table.

When merging multiple Update Sets, if several Update Sets have modified the same object, (for example: the Incident form), the most recent change will be the one moved to the new, merged Update Set.

Use an Update Set to migrate your code. When an Update Set is completed, you can transfer it to another instance to move customizations from development, through testing, and into production.

Batch update sets enable you to group update sets together so you can preview and commit them in bulk.

you can create a new Update Set or set an existing one as your current Update Set.All >

System Update Sets > Local Update Sets - new - fill form submit

When you have completed the configurations and compared local update sets to resolve conflicts, mark the update set as Complete.

In the Related Links section, select Export to XML.

The XML file downloads to your local computer.
Confirm the exported .xml update set file is saved to your local computer. The file name should begin with: **sys_remote_update_set_**

Retrieve

a. Navigate to All > System Update Sets > Retrieved Update Sets
b. Click Import Update Set from XML
c. Choose a file to upload
d. Upload the file

Preview

Commit

Business Rule

A Business Rule is configured to run when a record is displayed, inserted, updated, deleted, or when a table is queried.

- Before a record is saved to the database
- After a record is saved to the database
- Async (queued); client and server work independently so the client is not waiting for the server
- Display before the record is displayed

Business Rules execute on the **server side the table they are stored in is sys_script.**

ServiceNow Events

In ServiceNow, **Events** are system activities or occurrences that are triggered when specific conditions are met. These events are typically used to respond to changes in the system by initiating automated actions such as notifications, script executions, or logging. Events allow forreal-time monitoring and reaction to incidents, changes, or other activities in the ServiceNow environment.

1. *Event Generation:*
    - Events can be triggered automatically by system processes (e.g., a record update, incident creation, or workflow execution) or manually by scripts.
    - Common sources of event generation include:
        - Business rules.
        - Workflows and Flow Designer.
        - Scheduled jobs.
        - API calls.
2. *Event Registry:*
    - The **Event Registry** is where events are defined and configured in the system. Each event has:
        - **Name:** A unique identifier for the event (e.g., `incident.closed`, `change.requested`).
        - **Description:** Provides context about what the event represents.
        - **Queue:** Determines how the event is processed (e.g., default queue or a specific queue for high-priority events).
        - **Parameters:** Events can pass additional data (e.g., record ID, user details) to further customize actions.
3. *Event Queue:*
    - When an event is generated, it is placed in the **Event Queue**, where it waits to be processed.
    - The queue ensures that events are handled in the order they were generated, and it helps manage load by processing events asynchronously.
4. *Event Processing:*
    - Once an event is in the queue, it can trigger specific actions, such as:
        - **Notifications:** Sending emails, SMS, or push notifications based on the event.
        - **Scripts:** Running server-side scripts to perform actions like updating records or integrating with other systems.
        - **Logging:** Events can log information for auditing or monitoring purposes.

Common Event Triggers in ServiceNow

1. *Record-Based Events:*
    - Events are often triggered by changes to records. For example:
        - When an **incident is closed**, an event like `incident.closed` is generated.
        - When a **change request is created**, the `change.requested` event may trigger.

2. *Workflow and Flow Designer:*
    - Events can be triggered within **Workflows** or **Flow Designer** when specific conditions are met during the execution of a process (e.g., approval steps, task completions).
3. *Business Rules:*
    - Events are commonly triggered by **Business Rules**, which run when records are created, updated, or deleted. A business rule can be configured to trigger an event when certain conditions are met.
4. *Scheduled Events:*
    - **Scheduled jobs** can trigger events based on time-based conditions, such as daily reports, maintenance windows, or task reminders.

----------**END**----------