**Week – 1**

**Module 1 - ServiceNow Platform and Development Fundamentals.**

ServiceNow is a cloud-based platform founded in 2003 that helps large enterprises streamline IT delivery, enabling business users to solve problems independently.

**Highlights**

1. ServiceNow employs over 17,000 people globally and is recognized as a great workplace.
2. Founded in 2003, it became publicly traded in 2012 and was named a top innovative company in 2018.
3. Fred Luddy founded ServiceNow to address inefficiencies in traditional IT services and improve business interactions.
4. ServiceNow provides a cloud-based platform for IT and business workflows, enabling self-service for users.
5. Headquartered in Santa Clara, California, with global offices and data centers supporting the platform.
6. Targets mid to large enterprises, including major brands like Microsoft, Coca-Cola, and the NBA.
7. Led by CEO Bill McDermott, who previously expanded SAP’s market value significantly.
8. ServiceNow’s cloud-based model democratizes IT services, allowing users to resolve issues without relying on IT departments. This empowers business units and fosters efficiency.
9. The transition from Glidesoft to ServiceNow reflects a strategic evolution in addressing
10. IT delivery challenges, showcasing the importance of adaptability in tech companies.
11. Fred Luddy’s vision stemmed from his frustration with traditional IT, highlighting the need for platforms that prioritize user experience and solve real business problems.
12. The platform supports extensive customization, enabling businesses to create tailored solutions, reflecting the shift towards personalized software experiences.
13. With a global presence, ServiceNow leverages diverse markets and data centers, ensuring reliability and scalability for clients worldwide.
14. Recognition from Fortune and Glassdoor emphasizes ServiceNow’s strong workplace culture, which can attract top talent and drive innovation.
15. The diverse range of applications within ServiceNow allows businesses to select solutions that fit their unique needs, making it versatile and efficient for various industries.

**ServiceNow User Interface:**

* **Platform Architecture:** Introduction to ServiceNow's robust cloud-based architecture that integrates IaaS, PaaS, and SaaS, allowing for seamless application development and management.
* **User Interfaces:** Coverage of various interface options, including mobile apps and customizable service portals, enhancing user experience and accessibility.
* **Role-Based Access:** Explanation of how role-based access controls ensure security by limiting user access to sensitive data and functionalities.
* **Multi-Instance Architecture:** Insights into the platform's data segregation through its multi-instance architecture, ensuring high security and control.
* **Applications:** Discussion on the suite of applications, categorized into IT, Employee, Customer, and Creator workflows, supporting comprehensive business management.
* **Security and Backups:** Emphasis on the importance of regular backups and security protocols, supported by third-party certifications.
* **Personal Instance Setup:** Introduction to setting up a personal ServiceNow instance for hands-on learning and practical skills development.
* **Integration of Cloud Services:** ServiceNow's platform combines IaaS, PaaS, and SaaS, providing a flexible and secure environment for business operations.
* **Security through Role-Based Access:** Role-based access is vital for protecting sensitive information, ensuring users have appropriate access based on their roles.
* **Data Integrity and Security:** The multi-instance architecture and commitment to security certifications ensure robust data protection and integrity.

**List and Filter**

ServiceNow's list and filter functionalities offer users a robust way to view, sort, and analyse data across various database tables, significantly enhancing productivity and ease of use.

* ServiceNow uses lists to display database table contents.
* Filters allow users to customize their data views effectively.
* The dot list command provides quick access to any table.
* Users can create saved views and filters for future use.
* Grouping options help organize data more effectively.
* Personalization tools enable users to tailor their list views.
* Context menus enhance interaction with list data.

**Forms in ServiceNow**

* **Accessible Data Import Process**: The series is designed to make the data import process in ServiceNow understandable and accessible to users.
* **Importance of Staging Tables**: Staging tables are crucial intermediaries, simplifying the data import workflow between source and target entities.
* **Essential Terminology**: Understanding the terminology used in ServiceNow is vital for effectively managing data imports.
* **Automation of Staging Tables**: ServiceNow's automatic creation of staging tables reduces manual effort and minimizes potential errors in the import process.
* **Structured Data Management**: The import process is organized around three distinct data entities, which enhances clarity in data management.
* **Optimized Data Handling**: Scheduling data imports can improve data management and ensure timely updates within ServiceNow.

**Importing Data in ServiceNow**

* **Data Source**: The origin of the data to be imported.
* **Target Entity**: The final destination for the imported data within ServiceNow.
* **Import Set Table (Staging Table)**: An intermediary table automatically created by ServiceNow during the import process to facilitate data handling.

#### **The Import Process Overview**

* The import process in ServiceNow involves two primary data entities:
  + **Source Data Entity**: The original location of the data to be imported.
  + **Target Entity**: The destination within ServiceNow where the data will be stored.
* A critical component in this process is the **Import Set Table (Staging Table)**, which serves as an intermediary between the source and target entities, streamlining the import process.

#### **Understanding the Staging Table**

* The **Staging Table** is automatically generated by ServiceNow when an import is initiated, eliminating the need for manual creation.
* The terms **Staging Table** and **Import Set Table** will be used interchangeably throughout this series to simplify understanding.

#### **The Data Entities Involved**

* The import process includes three key data entities:
  + **Source Data**: The initial data set to be imported.
  + **Staging Table**: The temporary holding area for data during the import process.
  + **Target Data Store**: The final location within ServiceNow where the imported data is stored.

#### **Initiating the Import Process**

* The first step in the import process is to create a **Data Source** in ServiceNow. This step is foundational and will be explored in detail in the next note, setting the stage for further steps in the import process.

**Creating a Data Source in ServiceNow**

The process of **creating a data source** in ServiceNow as part of the data import workflow. This is a crucial step in understanding how data is transferred from external sources into ServiceNow’s environment. The fundamental entities involved in this process include the **source entity**, which holds the data, and the **target entity**, which is the destination within ServiceNow. An intermediary component, referred to as an **import set table** or **staging table**, is also vital in managing the data during import. This chapter will outline the steps necessary to define a data source, detailing the parameters and configurations required to facilitate effective data integration.

## **Creating a Data Source**

* **Definition of Data Source**: A data source in ServiceNow is a record that contains parameters essential for identifying the type of source data entity and how to connect to it.
* **Parameters to Define**:
  + Source location information.
  + Data to import (full dataset or a subset).
  + Naming conventions for the staging table.

### Accessing the Data Source Table

* The data source records are stored in the **sys\_data\_source** table.
* To view existing records:
  + Use the Application Navigator: Enter sys\_data\_source.list to retrieve records.
  + Navigate through the System Import Sets > Administration > Data Sources.

### Creating a New Data Source

* To create a data source, click the **New** button in the list view of the data source table.
* **Naming the Data Source**: Assign a name (e.g., “test import”) for easy identification.
* **Staging Table Configuration**:
  + Define the label (e.g., “test import”) and the actual table name (e.g., u\_test\_import).

## **Types of Data Sources**

* ServiceNow supports various data source types, including:
  + **File**: Commonly used for importing data from files like Excel or CSV.
  + **JDBC**: For connecting to relational databases (e.g., Oracle, SQL Server).
  + **Other Types**: LDAP, OIDC, REST, or custom scripts.

### File-Based Data Imports

* When selecting **file** as the type, specify:
  + File format (e.g., CSV, Excel).
  + Sheet number if applicable.
  + Header rows for proper field creation in the staging table.
  + **File Retrieval Method**: Options include various file transfer protocols or attachments.

## Setting Up a JDBC Data Source

* When selecting **JDBC**, additional parameters are required:
  + Database type (MySQL, Oracle, etc.).
  + Database name and port.
  + Connection credentials (username and password).
  + SQL statement or table name to define the data to be imported.

## Attaching the Data File

* After defining the data source, attach the data file (e.g., an Excel spreadsheet) to establish access to the data.
* The header row in the file will dictate the fields created within the staging table during data import.

**Understanding Import Sets in ServiceNow**

In this chapter, we delve into the critical concept of **Import Sets** within the ServiceNow platform, focusing particularly on the **staging table** created during the import process. This discussion is part of a broader series aimed at elucidating the steps involved in importing data into ServiceNow. Understanding how the staging table functions is vital as it serves as a temporary holding area for data before it is finalized in the target table. Key vocabulary terms include **data source**, **staging table**, **import set table**, and **import set row table**, all of which will be explored in depth.

## Data Source Creation

* **Data Source Record**: In the previous note, a **data source record** was created to specify the origin of the data to be imported.
* **Attributes**: This record includes essential parameters for the staging table, such as its **name** and **label**. For example, the staging table was designated as U\_test\_import with the label “Test Import.”
* **Test Drive**: The next logical step is to validate the data source by running a test import to confirm the connection and the creation of the staging table.

## Staging Table Creation

1. **Automatic Creation**: When the import is first executed, ServiceNow checks if the staging table exists. If not, it creates one based on the data source parameters.
2. **Load Data**: After confirming the table’s creation, ServiceNow pulls data from the specified source and loads it into the staging table.
3. **Error Handling**: If an attempt is made to view the table before running an import, an error message indicates that the table does not exist, reinforcing the need to execute the import first.

## Running the Import

1. **Testing Data Load**: The data source allows testing of the import functionality, offering options to load a specific number of records (in this case, five from an Excel file).
2. **Successful Completion**: Upon running the import, a confirmation message appears, detailing how many records were processed and inserted into the staging table.
3. **Verification**: After the import, the staging table now reflects the five records successfully loaded.

## Staging Table Structure

* **Table Configuration**: The structure of the staging table can be analyzed through the **hamburger menu**, which provides insights into the custom columns created based on the data source.
* **Column Identification**: Custom columns, labeled with a red X, indicate that these are not default fields but are generated from the imported data.

## Managing Import Sets

1. **Import Set Table**: ServiceNow maintains an out-of-the-box **import set table** (sys\_import\_set), which tracks each import run.
2. **Unique Identifiers**: Each time an import is executed, a new record is created in the import set table, referencing the corresponding staging table entries.
3. **Group Management**: The ability to manage records within the staging table is essential, as multiple import runs can lead to redundant entries. The import set table helps keep track of which records belong to which import run.

**ServiceNow Transform Maps & Field MapsIntroduction**

In this chapter, we delve into the process of importing data into ServiceNow, focusing particularly on Transform Maps and Field Maps. The significance of this topic lies in its critical role in ensuring that data is accurately moved from a staging area to a target destination within the ServiceNow platform. Understanding these concepts is essential for effective data management and integration in ServiceNow.

Key Vocabulary:

* Transform Map: A record that groups together multiple Field Maps for a particular import process.
* Field Map: A mapping of individual fields from the staging table to the target table.
* Staging Table: An intermediary table that temporarily holds imported data before it is processed.
* Coalesce: A method to identify unique records to prevent duplicates during data import.

Creating a Data Source

* The first step in the import process is establishing a Data Source, which informs ServiceNow about the origin of the data, the type of data, and how to connect to it.
* A staging table is created based on the data source, allowing access to the data before it reaches the final destination.

Testing the Data Source

* After creating the data source, it is essential to test the connection and ensure that the staging table is set up correctly, displaying imported data as intended.

Transform Maps and Field Maps

* The next critical phase involves creating Transform Maps and Field Maps to facilitate the movement of data:
  + Field Maps specify how individual fields in the staging table correspond to fields in the target table, ensuring accurate data transfer.
  + Each mapping between fields is stored as a record in the Field Map table (CIS\_transform\_entry), with a separate record for each mapping.
  + Transform Maps act as a grouping mechanism for these field mappings, stored in the Transform Maps table (CIS\_transform\_map).

Setting Up the Target Table

* The target table can be an out-of-the-box ServiceNow table or a custom table created for specific purposes.
* In this demonstration, a custom table named “my\_table” was created, which includes fields for the imported data (username, address, city, state, zip code).

Building the Transform Map

* The process begins by creating a new Transform Map record, specifying the source (staging table) and target (final destination table).
* After saving the transform map, the next step involves setting up the corresponding Field Maps using the mapping assistant tool, which streamlines the mapping process.

Coalesce Field

* The Coalesce field is crucial for identifying unique records during the import process to prevent duplicates.
* In this example, the “name” field was chosen as the coalesce field, but it is emphasized that a unique identifier would be more effective in real-world scenarios.

Verifying Records

* To ensure accuracy, it is essential to check the Field Maps and Transform Maps tables post-setup to verify that all records are correctly created.

**ServiceNow Incident Management Tutorial and Task Administration**

Tasks in ServiceNow are defined as records that capture items of work needing completion. Users can access the task table by entering “task.list” in the application navigator, which displays all existing task records.

* **Common Attributes of Tasks:**
  + Description: A brief explanation of the task.
  + Status: Current state of the task.
  + Due Date: Deadline for task completion.
  + Assigned To: Individual responsible for the task.

ServiceNow categorizes tasks into various types, with change requests, incidents, and problems being the most prevalent. These are structured in a hierarchical database design where each type inherits attributes from the task table while adding specific fields relevant to its function.

* Key Point:
  + The task table serves as a foundation, while extended tables hold specific task attributes.

**Business Value of Task Management**

The capabilities provided by ServiceNow’s task management can significantly enhance organizational efficiency. By defining and managing tasks, organizations can establish repeatable processes for common work items.

* Core Functionalities:
  + Assignment Rules: Automatically assign tasks to appropriate users or groups.
  + Approval Processes: Manage task approvals efficiently.
  + Service Level Agreements (SLAs): Monitor task completion within specified timeframes.
  + Inactivity Monitors: Notify stakeholders when tasks remain untouched for long periods.
  + Workflows: Create tailored workflows for tasks meeting certain criteria.

The initial step in task management involves assigning tasks to the right personnel by populating the assigned to and assignment group fields. This requires a well-maintained user and group table, ensuring proper task allocation.

**Creating and Managing Assignment Rules**

Assignment rules dictate how tasks are assigned in ServiceNow. These rules are recorded in the assignment rule table and allow multiple conditions to be set for assigning tasks.

* Key Process:
  + Create a new assignment rule that specifies conditions such as task category (e.g., hardware incident) and target assignment (user and group).
  + The execution order of rules determines how they are processed, with lower numbers being prioritized.

**Example of Creating an Assignment Rule**

An example discussed is creating an assignment rule for hardware incidents, automatically assigning them to a designated group and user.

* Steps:
  + Define conditions for the rule.
  + Assign the appropriate user and group.
  + Set the execution order.

**Task Completion Perspective**

Users working on tasks utilize the service desk application to access their assigned tasks or those of their groups. The My Work and My Group’s Work options allow users to manage tasks effectively.

* Collaboration Tools:
  + User Presence: Shows active users viewing the same record.
  + Real-time Editing: Reflects changes made by others immediately.

These tools enhance teamwork and expedite task resolution by facilitating communication among stakeholders.

**Visual Task Boards**

ServiceNow offers visual task boards as a graphic alternative to lists, allowing users to organize and manage tasks visually.

* Components:
  + Cards: Represent individual tasks.
  + Lanes: Group tasks based on defined attributes (e.g., categories).

**Types of Task Boards**

* Guided Boards: Created from lists with predefined values.
* Flexible Boards: Built from lists without predefined values, allowing for customization.
* Freeform Boards: Personalized boards for managing various tasks without direct ties to existing records.

**ServiceNow Reporting**

**Understanding ServiceNow’s Reporting Capabilities**

* **Database Structure:** The foundation of reporting in ServiceNow lies within the sys\_report table, a system table that stores records for all existing reports.
* **Supportive Tables:** There are four additional tables that facilitate reporting:
  + Report Source Table: Stores saved queries for report data retrieval.
  + Scheduled Email Reports Table: Manages automatic report execution and emailing.
  + Report Users and Groups Table: Allows sharing of reports with specific users or groups.
  + Dashboard Table: Integrates reports into user-defined dashboards for broader data presentation.

Key Facts:

* The sys\_report table contains 156 fields that detail report attributes.
* There are over 23 types of reports available in ServiceNow, such as pie charts, bar charts, and heat maps.

**Creating and Managing Reports**

* Creating Reports: Users can create reports via three methods:
  + Using the Reports Application.
  + Utilizing ServiceNow Studio.
  + Starting from an existing list view.
* **Fields in the sys\_report Table:**
  + sys\_id: Unique identifier for records.
  + Title: Displays the report title.
  + Source Type: Indicates if data comes from a table or a saved query.
  + Table Field: Specifies the data source table.
  + Filter Field: Defines filtering conditions for report data.
  + Type Field: Determines the report visualization type.

**Key Statistics:**

* There are 652 reports configured in the sample ServiceNow instance presented.

**Scheduling and Sharing Reports**

* Scheduling Reports: Users can automate report distribution using the sys\_auto\_report table. Users can specify:
  + Report execution frequency (daily, weekly, monthly).
  + Recipients of the report.
* Sharing Reports: The sys\_report\_users\_groups table enables sharing reports with individuals or groups, allowing for on-demand access.

Key Example:

* A report titled “Users by Department” can be scheduled for monthly email distribution to specific users and groups.

Integrating Reports into Dashboards

* Dashboards: ServiceNow allows for the consolidation of multiple reports into a single dashboard view. This integration involves creating relationships between various tables, culminating in the pa\_dashboard table.

Example Process:

* Adding a report to a dashboard can be done by selecting the report and specifying the desired dashboard and location.

**What is Low Code No Code Development?**

**Traditional Software Development**

* The journey begins in the realm of **traditional software development**, characterized by a lengthy and complex process.
* The Savvy Business Person identifies an opportunity for digital transformation but faces barriers such as the need for a server, database, and coding.
* A cycle of requirements gathering, feedback, and iterative design often results in misalignment between business needs and IT deliverables.

**The Emergence of Low Code No Code**

* The narrative shifts as Low Code No Code enters the scene, aiming to eliminate the barriers represented by The Wall.
* LCNC enables Savvy Business People to tackle business challenges independently, utilizing intuitive tools without requiring extensive technical knowledge.
* The approach focuses on **empowering users**, allowing them to build and manage applications through simplified interfaces.

**Tools and Resources**

* **ServiceNow** is highlighted as a prime example of a company offering LCNC capabilities.
* **App Engine Studio**: Guides users in building applications without deep technical know-how.
* **UI Builder**: Provides a drag-and-drop interface for creating portals and workspaces.
* **Flow Designer**: Automates workflows using natural language processing.
* Other companies in the LCNC space include **Microsoft PowerApps**, **Zoho**, **Appian**, and **Salesforce**.