Creating a Data Source in ServiceNow

What is Data Source?

A data source is a record that stores parameters needed to connect to and import data from various sources.

Another definition:

A data source defines the location and format of the source data used in data import processes.

Steps to create a Data Source:

- **Specify Data Source Type:** Select the appropriate type of data source based on the source data's location and format (e.g., CSV, Excel, database).
- **Define Location and Format:** Provide details about the location of the source data (e.g., file path, URL) and its format.
- **Map Fields:** Map the fields in the source data to the corresponding fields in the ServiceNow staging table.
- **Attach File (Optional):** If the source data is a file, attach it to the data source.
- **Create Staging Table:** ServiceNow automatically creates a staging table (import set table) to temporarily store the imported data.

Understanding Import Sets in ServiceNow

- **Staging Table Creation:** ServiceNow automatically creates a staging table based on the parameters specified in the data source record. The speaker also mentioned that the staging table is an intermediary table between the source entity and the target entity. The staging table is an intermediary storage for data before it's loaded into the target table.
- **Import Set Table:** The import set table tracks import runs and associate's records with specific runs. Import sets are staging tables used to temporarily store data during the import process in ServiceNow. They play a crucial role in ensuring data integrity and tracking import runs.
- **Testing Data Source:** It's essential to test the data source to ensure the connection works and data is loaded correctly into the staging table.

ServiceNow Transform Maps & Field Maps

Field Maps: Field maps specify how individual fields in the staging table should be mapped to fields in the target table. This mapping is crucial for ensuring that data is imported accurately and consistently.

- **Mapping:** Field maps specify the exact correlation between fields in the two tables.
- **Data Types:** It's essential to ensure that the data types of the source and target fields are compatible.
- **Customizations:** Field maps can be customized to handle various data transformation scenarios, such as formatting, calculations, etc.

Transform Maps: Transform maps group field maps together and define the overall import process. To create a transform map, you need to specify the source table (staging table), target table, and field mappings.

- **Grouping:** Transform maps bundle related field maps into a single unit.
- **Process Definition:** They define the sequence of operations and the conditions that govern the import process.
- **Customization:** Transform maps can be customized to handle complex import scenarios, such as conditional processing, error handling, and data validation.

Coalesce Fields: Coalesce fields are used to identify matching records during the import process to prevent duplicates. It prevents the duplication of records when importing data from multiple sources.

After creating transform maps and field maps, it's important to test the import to ensure data is moved correctly.

ServiceNow Incident Management Tutorial and Task Administration

Task Table: Tasks are stored in the task table, which is a core component of the ServiceNow platform. Common task types include change requests, incidents, and problems.

Assigning Tasks:

- Tasks can be assigned to users or groups using the assigned to and assignment group fields.
- Assignment rules can automate the assignment process based on specific conditions.

Visual task boards provide a visual representation of tasks, allowing for easy management and tracking.

There are three types of task boards: **guided, flexible,** and **freeform**.

Guided boards use predefined values for lanes, while flexible boards allow for custom lane definitions.

Freeform boards are not tied to specific task records and can be used for general work management.

ServiceNow Reporting Tutorial

Data Model: The underlying data model for reports includes the sys_report table, report_source table, scheduled_email_of_reports table, report_users_and_groups table, and dashboard table.

ServiceNow supports various report types, including list reports, chart reports, pivot table reports, and KPI reports.

Report Creation: Reports can be created using the report designer or by starting from a data source.

Field Mapping: Field mapping defines how data from the source is mapped to fields in the report.

Filtering and Grouping: Filtering and grouping allow users to refine the data displayed in reports.

Formatting: Reports can be customized with various formatting options to enhance readability and presentation.

Sharing and Publishing: Reports can be scheduled for automatic execution and emailed to users, shared with individual users or groups, and added to dashboards for easy access and visualization.

Other than this we can:

- We can create and edit reports using the reports application, ServiceNow Studio, or from an existing list view.
- We can schedule reports to be emailed to users on a recurring basis.
- We can share reports with other users or groups of users and add reports to dashboards.

What is Low Code No Code Development?

Low-Code/No-Code (LCNC) platforms have transformed the landscape of software development by empowering individuals with limited technical expertise to create applications. These platforms provide a visual interface and pre-built components, allowing users to assemble applications without writing extensive code.

How Low-Code/No-Code Works:

- **Visual Interface**- LCNC platforms offer a drag-and-drop interface where users can visually design the layout and components of their applications.
- **Pre-built Components**- These platforms provide a library of pre-built components, such as forms, buttons, charts, and databases, which can be easily integrated into applications.
- **Configuration**- Users can configure these components to suit their specific needs, defining their behavior, appearance, and data connections.
- **Integration-** LCNC platforms often allow for integration with existing systems and APIs, enabling seamless communication and data exchange.

Benefits of LCNC:

- **Faster Development Time**: LCNC platforms significantly reduce development time by eliminating the need for extensive coding.
- **Reduced Costs:** By accelerating development and minimizing the need for specialized technical resources, LCNC platforms can lower development costs.
- **Increased Agility**: LCNC platforms enable organizations to respond quickly to changing business needs by rapidly creating and modifying applications.

Limitations of LCNC:

While LCNC platforms offer numerous advantages, they may have limitations in terms of customization and flexibility. Due to their general-purpose nature, LCNC platforms may not be able to handle highly complex or specialized requirements.

Career Opportunities:

- Citizen Developer
- Low-Code/No-Code Developer
- Low-Code/No-Code Architect