Automated Network Request Management in ServiceNow

Data Architecture

1. Introduction

Data architecture plays a critical role in the Automated Network Request Management system by defining how network-related information is structured, stored, and managed within ServiceNow. A well-designed data architecture ensures data consistency, scalability, and seamless integration with automation workflows. This phase focuses on creating a custom table, defining fields, and configuring forms to support backend processing.

2. Creation of Network Database Table

Objective

To create a centralized table to store and manage all network request-related data.

Procedure

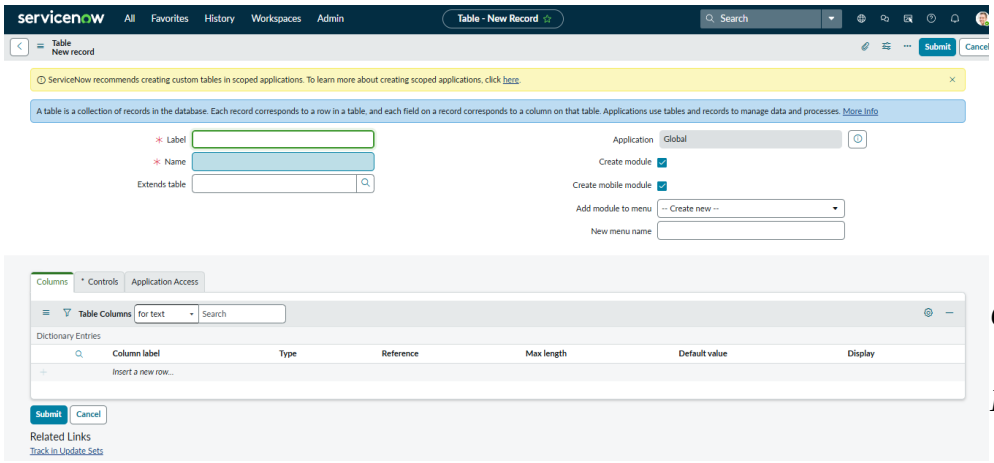


Figure 2.1: Creation of Network Database Table in ServiceNow

- Navigate to **System Definition → Tables**.
- Click **New** to create a new table.
- Enter the following details:

- **Name:** Network Database
- **Label:** u_network_database

- Click **Submit** to create the table.

This table serves as the primary backend repository for network request records.

3. Creation of Fields

Objective

To define structured fields (columns) that store request-specific information in the Network Database table.

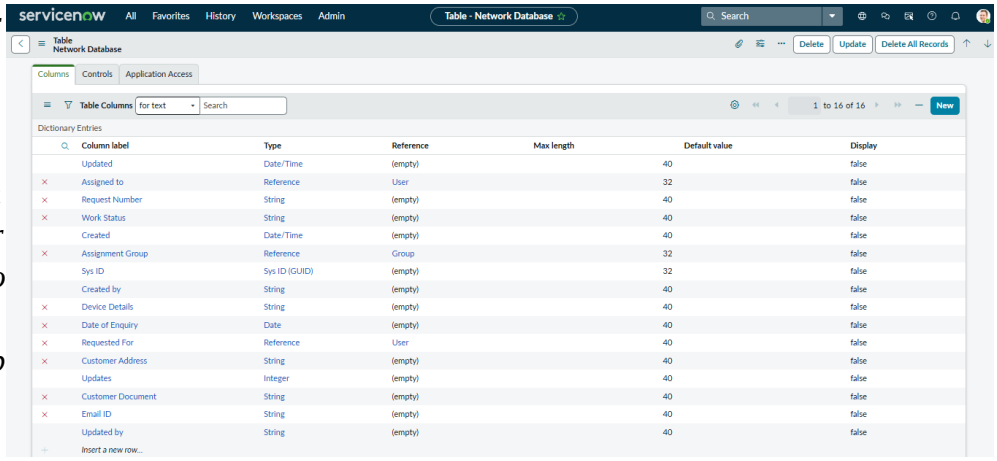
Procedure

- Open **System Definition → Tables**.
- Search and select **Network Database**.
- Navigate to the **Columns** related list.
- Click **New** to add a field.
- Define required field attributes and submit.

4. Field Properties Configuration

Each field is configured using appropriate properties to ensure correct data storage and validation.

Figure 4.1: Columns Defined for Network Datab



ase Table

Field Attributes

- **Column Label:** User-friendly field name
- **Column Name:** Auto-generated internal field name
- **Data Type:**
    - String
    - Integer
    - Choice
    - Reference
- **Mandatory:** Enabled for critical data
- **Default Value:** Configured when required
- **Read-Only:** Used for system-generated or calculated fields

After configuration, fields are saved and added to the table schema.

5. Adding Fields to Forms

Objective

To make backend fields accessible to users through forms.

Procedure

- Navigate to **System UI → Forms**.
- Open the required form.
- Launch **Form Designer**.
- Drag the newly created fields onto the form layout.
- Save or publish the form.

6. Testing and Validation

- Create a new record in the Network Database table.
- Verify field visibility and behavior.
- Validate mandatory, read-only, and default value configurations.

7. Best Practices Followed

- Proper naming conventions for fields
- Selection of appropriate data types
- Use of UI Policies and Client Scripts where required
- Consistent schema design for scalability

8. Conclusion

The data architecture for the Automated Network Request Management system establishes a robust foundation for backend operations. By creating a dedicated table with well-defined fields and validations, the system ensures reliable data storage, seamless automation, and future scalability.