

# Calculating Family Expenses Using ServiceNow

**Team Id:** NM2025TMID13341

Team Leader: R.Nivetha

Team Member 1: S.Priyadharshini

TeamMember 2 : D.Sneha

Team Member 3 : K.Sharmila

## Abstract

This project focuses on designing and implementing a **Family Expense Management System** using the **ServiceNow platform**. The system enables recording, tracking, and analyzing family expenditures in a structured and automated way. Key features include creating custom tables, defining columns, configuring forms, applying business rules for automation, and establishing relationships between daily and overall expenses.

## Introduction

Managing household expenses is often challenging without a proper tracking system. ServiceNow provides a platform to create customized applications that can automate this process. By building a dedicated **Family Expenses application**, we can:

- Record daily expenditures
- Ensure mandatory expense details are filled
- Automate data handling with business rules
- Maintain relationships between daily expenses and total family expenses

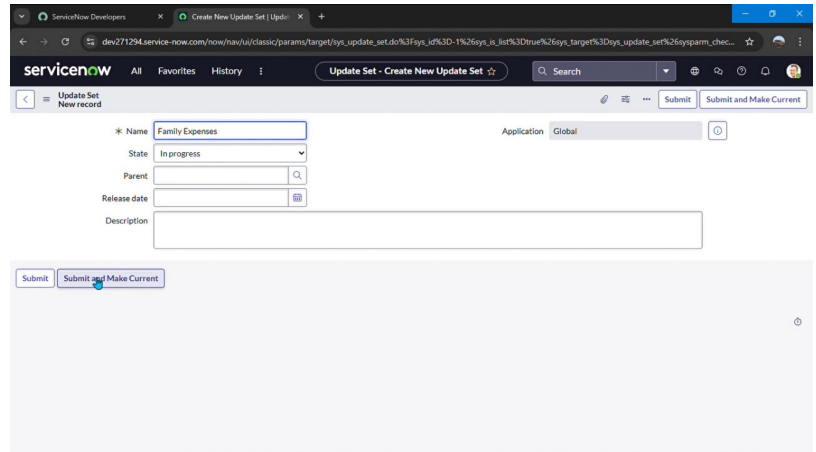
This project demonstrates how to leverage ServiceNow's low-code platform to solve a real-life problem with minimal effort.

## Methodology

### Step 1: Creation of Update Set

Update sets in ServiceNow act as containers to store configuration changes.

- Navigate to **All > Local Update set**.
- Click **New**.
- Enter **Name:** Family Expenses.
- Submit and mark it as **Current**.

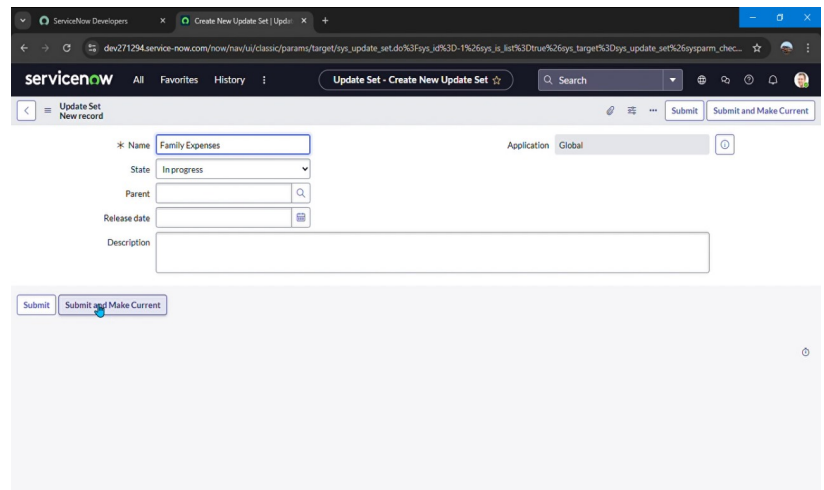


*Why?* → To track and manage all changes related to the Family Expenses application in one place.

### Step 2: Creation of Family Expenses Table

Tables are the backbone of ServiceNow applications.

- Navigate to **All > Tables > New**.
- Enter details:
- **Label:** Family Expenses
- **Name:** Auto-generated
- **Menu Name:** Family Expenditure
- Save the table.



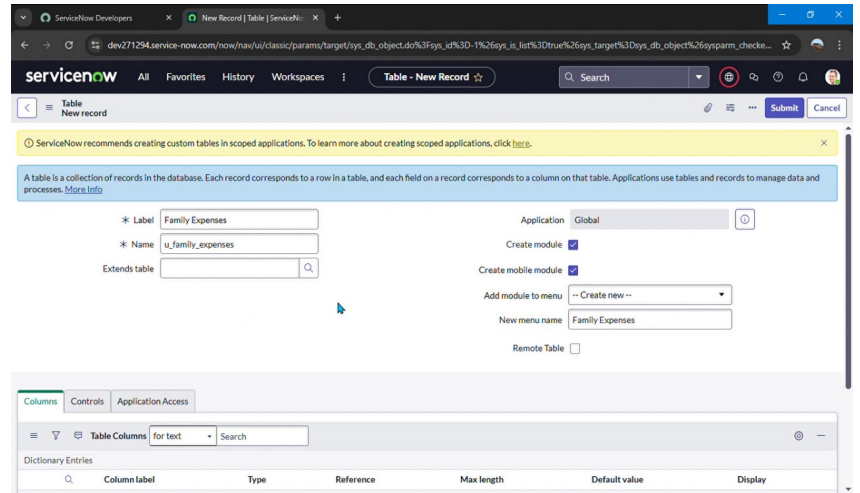
*Why?* → This table stores the overall expense records of the family.

## Step 3: Adding Columns (Fields)

Columns (fields) define the structure of the table. For Family Expenses, the following fields were created:

- **Number** → String (Unique record identifier)
- **Date** → Date (Expense date)
- **Amount** → Integer (Total expense amount)
- **Expense Details** → String (Details/description, max 800 characters)

*Why?* → To capture complete information about each expense entry.



The screenshot shows the 'Table - New Record' configuration page in ServiceNow. The 'Label' field is set to 'Family Expenses' and the 'Name' field is set to 'u\_family\_expenses'. The 'Application' is set to 'Global'. The 'Create module' checkbox is checked. The 'Add module to menu' dropdown is set to '-- Create new --'. The 'New menu name' is set to 'Family Expenses'. The 'Remote Table' checkbox is unchecked. Below the configuration fields, there is a 'Columns' tab with a table showing the columns for the table.

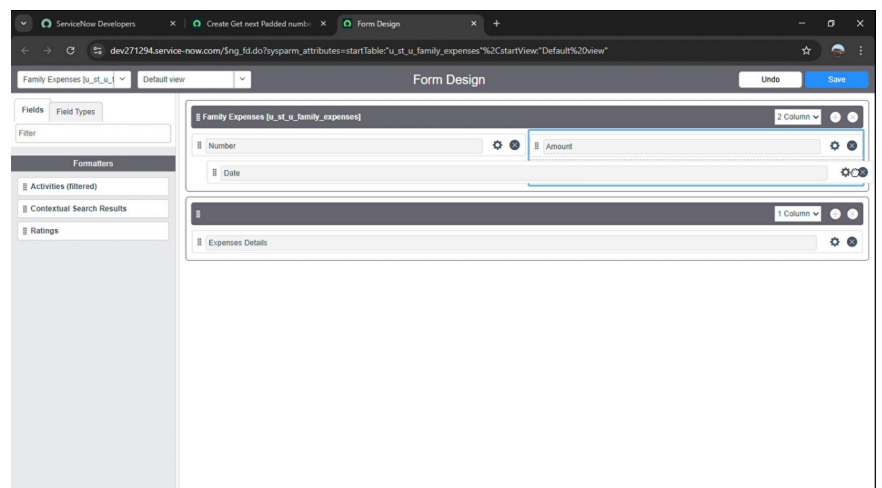
Column label	Type	Reference	Max length	Default value	Display
Number	String				
Date	Date				
Amount	Integer				
Expense Details	String		800		

## Step 4: Configuring the Form

Forms provide the interface for users to enter data.

- Search **Family Expenses** in the application navigator.
- Click **New** → Choose **Form Design**.
- Rearrange fields for better readability.
- Make **Date** and **Amount** mandatory fields.
- Save the form.

*Why?* → Mandatory fields prevent incomplete data entry, ensuring accuracy.

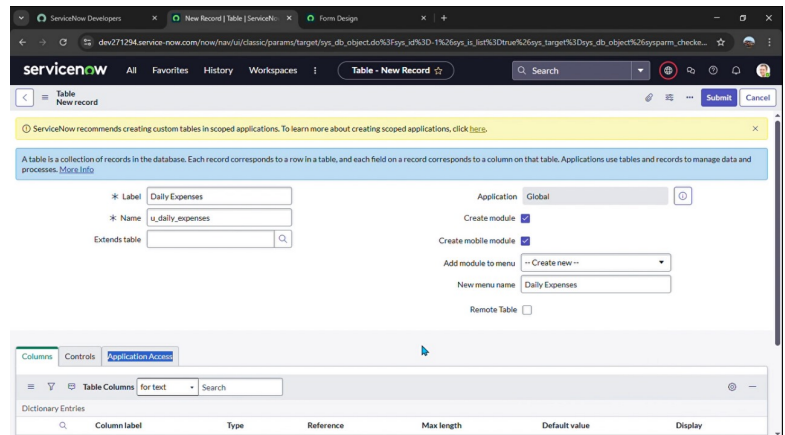


The screenshot shows the 'Form Design' interface in ServiceNow. The form is titled 'Family Expenses [u\_st\_u\_family\_expenses]'. The form contains four fields: 'Number', 'Date', 'Amount', and 'Expense Details'. The 'Number' field is a text field. The 'Date' field is a date field. The 'Amount' field is a numeric field. The 'Expense Details' field is a text area. The form is configured with a 'Default view' and a 'Form Design' tab. The 'Form Design' tab shows the form layout with fields arranged in a grid. The 'Number' field is in the first row, 'Date' is in the second row, 'Amount' is in the third row, and 'Expense Details' is in the fourth row. The 'Form Design' tab also shows a 'Fields' list on the left side of the form.

## Step 5: Creation of Daily Expenses Table

Daily tracking is important for detailed financial monitoring.

- Navigate to **All > Tables > New**.
- Enter details:
- **Label:** Daily Expenses
- **Name:** Auto-generated
- Save the table.



The screenshot shows the 'Table - New Record' form in ServiceNow. The 'Label' field is set to 'Daily Expenses' and the 'Name' field is set to 'u\_daily\_expenses'. The 'Application' is set to 'Global'. The 'Create module' checkbox is checked. The 'Add module to menu' dropdown is set to 'Create new...'. The 'New menu name' is set to 'Daily Expenses'. The 'Remote Table' checkbox is unchecked. Below the form, there is a table with columns: Column label, Type, Reference, Max length, Default value, and Display. The table is currently empty.

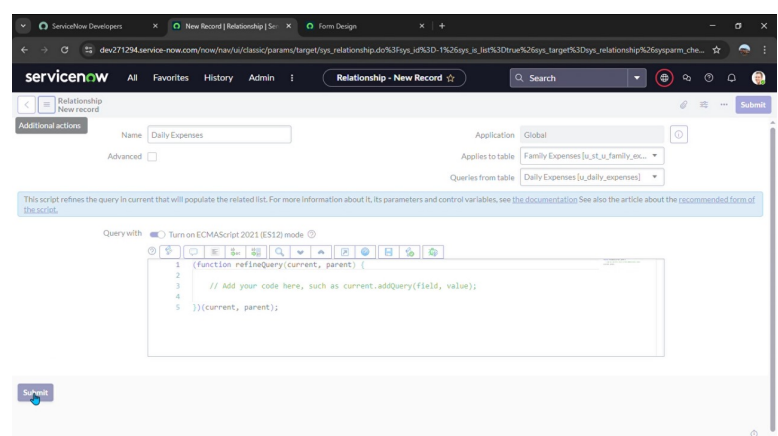
*Why?* → This table captures each individual daily transaction, which can later be linked to the Family Expenses table.

## Step 6: Business Rules Implementation

Business rules automate background processes in ServiceNow.

Navigate to **All > Business Rules > New**.

- Enter details:
- **Name:** Family Expenses BR
- **Table:** Daily Expenses
- **When to Run:** Insert & Update
- Add logic (e.g., automatically update Family Expenses when Daily Expenses are entered).
- Save the rule.



The screenshot shows the 'Relationship - New Record' form in ServiceNow. The 'Name' field is set to 'Daily Expenses'. The 'Application' is set to 'Global'. The 'Applies to table' dropdown is set to 'Family Expenses [u\_st\_u\_family\_expenses]'. The 'Queries from table' dropdown is set to 'Daily Expenses [u\_daily\_expenses]'. Below the form, there is a text area for the script. The script is as follows:

```

1 function refineQuery(current, parent) {
2   // Add your code here, such as current.addQuery(field, value);
3   // Add your code here, such as current.addQuery(field, value);
4   // Add your code here, such as current.addQuery(field, value);
5 }
  
```

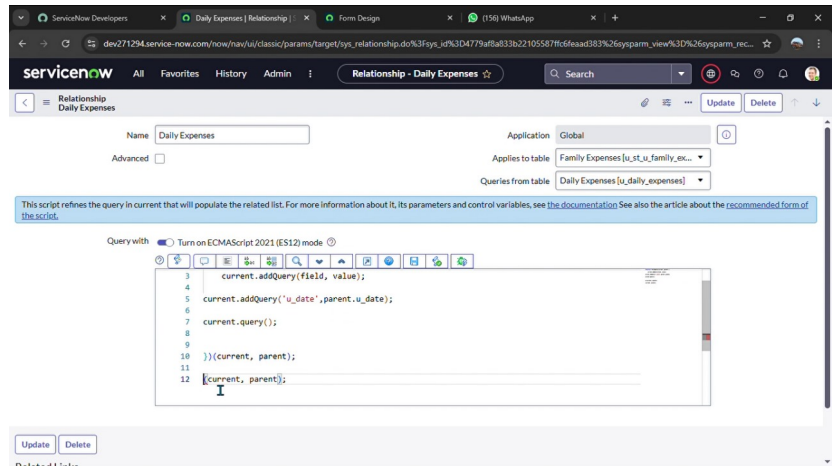
*Why?* → Automation reduces manual work and ensures that updates to Daily Expenses reflect in Family Expenses automatically.

## Step 7: Configuring Relationships

Relationships link different tables to make data meaningful.

Navigate to **All > Relationship**.

- Open **Daily Expenses**.
- Under **Applies to**, select **Family Expenses**.
- Add relationship code and update.



*Why?* → Establishes a parent-child relationship between Family Expenses and Daily Expenses, making reports and analysis easier.

## Results

A **Family Expenses** table was successfully created.

Users can record **daily expenses** with details like date, amount, and description.

**Forms** are configured with mandatory fields to ensure data completeness.

**Business Rules** automate updates between daily and overall expense records.

**Relationships** enable linking of data for better reporting.

## Applications

- Personal household expense tracking
- Budget planning for families

Can be extended for **small businesses** or **clubs/associations** to track group expenses

## Conclusion

The Family Expense Management System developed in ServiceNow provides a structured and automated way of managing household finances. By leveraging ServiceNow features such as **update sets, tables, forms, business rules, and relationships**, the project ensures accurate expense recording, efficient data entry, and improved financial tracking.

This project demonstrates how a simple real-world problem can be solved using the capabilities of a cloud-based low-code platform.