

CALCULATING FAMILY EXPENSES USING  
SERVICE NOW

NAAN MUDALVAN PROJECT

TEAM ID:

NM2025TMID17769

Team Leader : SAJIN S

Team member : SELVA HARINI S

Team member : SANTHIYA D

Team member : SANTHOSH KUMAR M

ST.JOSEPH'S COLLEGE (ARTS &  
SCIENCE)

## INDEX

1. Introduction
2. Abstract
3. Problem statement
4. Solution
5. Practical use
6. Knowledge gained
7. Milestone 1: Setting up service now instance
8. Milestone 2: Creation of new update set
9. Milestone 3: Creation of table family expenses
10. Milestone 4: Creation of table daily expenses
11. Milestone 5: Creation of relationship
12. Milestone 6: Configuring Related list on family expenses
13. Milestone 7: Creation of Business rules
14. Milestone 8: Configure the relationship
15. Conclusion

# INTRODUCTION

Managing family expenses is one of the most important aspects of financial planning, yet it is often overlooked or handled in unstructured ways like handwritten notes or spreadsheets.

Families frequently face challenges such as overspending, lack of budget control, and difficulty in analyzing where their money goes. To address these issues, technology can play a vital role by providing structured, automated, and user-friendly solutions.

This project, "Calculating Family Expenses Using ServiceNow," aims to create a digital system that helps track, categorize, and manage family expenses efficiently.

ServiceNow, known for its robust workflow automation and application development capabilities, provides an excellent platform for building such a system. By leveraging ServiceNow, this project transforms expense management into a more streamlined process with features like expense categorization, daily tracking, budget limits, reporting, and automated business rules.

The system will not only simplify financial management for families but also provide real-time insights into spending patterns, allowing better decision-making. With its scalability and flexibility, the solution can be expanded to suit different family structures and even adapted for small business expense management. Ultimately, the project showcases how a powerful enterprise platform like ServiceNow can be applied beyond IT workflows to solve everyday problems in an innovative and practical way.

## ABSTRACT

Expense management plays a crucial role in maintaining financial stability within families, yet traditional methods such as manual tracking or spreadsheets often lead to errors, lack of visibility, and poor decision-making. To address this challenge, this project focuses on designing and implementing a Family Expense Management System using ServiceNow.

The application leverages ServiceNow's powerful low-code development environment to create structured tables, relationships, and automated workflows for recording and analyzing expenses. Key features include categorizing expenses (such as food, utilities, and transport), maintaining family member details, linking daily transactions, and applying business rules to validate data and automate calculations. By configuring related lists and creating meaningful reports, the system provides users with a clear overview of their financial habits and budget limits.

The proposed solution not only simplifies day-to-day expense tracking but also provides real-time insights into spending patterns, enabling families to make smarter financial decisions. Furthermore, the project demonstrates the versatility of ServiceNow beyond traditional IT service management, showcasing its potential in solving practical, real-world problems.

## PROBLEM STATEMENT

Managing household expenses is often a difficult and time-consuming task for families. Most families rely on manual methods such as notebooks, receipts, or spreadsheets to track their daily spending. These methods come with several challenges:

- Lack of real-time tracking of expenses.
- Difficulty in categorizing and consolidating expenses for food, utilities, rent, and transportation.
- Limited ability to analyze spending patterns or generate reports.
- High chances of errors due to manual data entry.
- No automation to alert families about overspending or exceeding budgets.

As a result, families often lose visibility into their financial flow, making it harder to control budgets or make informed financial decisions. This creates the need for a systematic, automated, and user-friendly solution to manage and calculate family expenses efficiently.

## SOLUTION

The proposed solution is to build a Family Expense Management System on the ServiceNow platform. ServiceNow, being a robust low- code/no-code platform, provides all the tools required to create structured applications without needing extensive programming knowledge.

The solution involves:

- Creating custom tables to store family member details and daily expenses
- Defining relationships between family members and their respective expenses for easy tracking.
- Configuring related lists so that expenses linked to each member can be viewed in one place.
- Implementing business rules to automate calculations, entries, and ensure data accuracy.
- Generating reports and dashboards to visualize monthly/annual spending and highlight budget deviations.
- Using update sets to track and migrate customizations, proper version control, ensuring

## PRACTICAL USE

The Family Expense Management System built on ServiceNow helps families track, categorize, and analyze their expenses in a structured way. It simplifies budgeting, reduces errors from manual tracking, and provides real-time insights into spending patterns. Families can use it to set limits, monitor monthly expenses, and generate reports for better financial decisions. Beyond households, the same system can be adapted for small businesses to manage cash flow and daily transactions effectively, proving the versatility of ServiceNow in solving practical, non-IT problems.

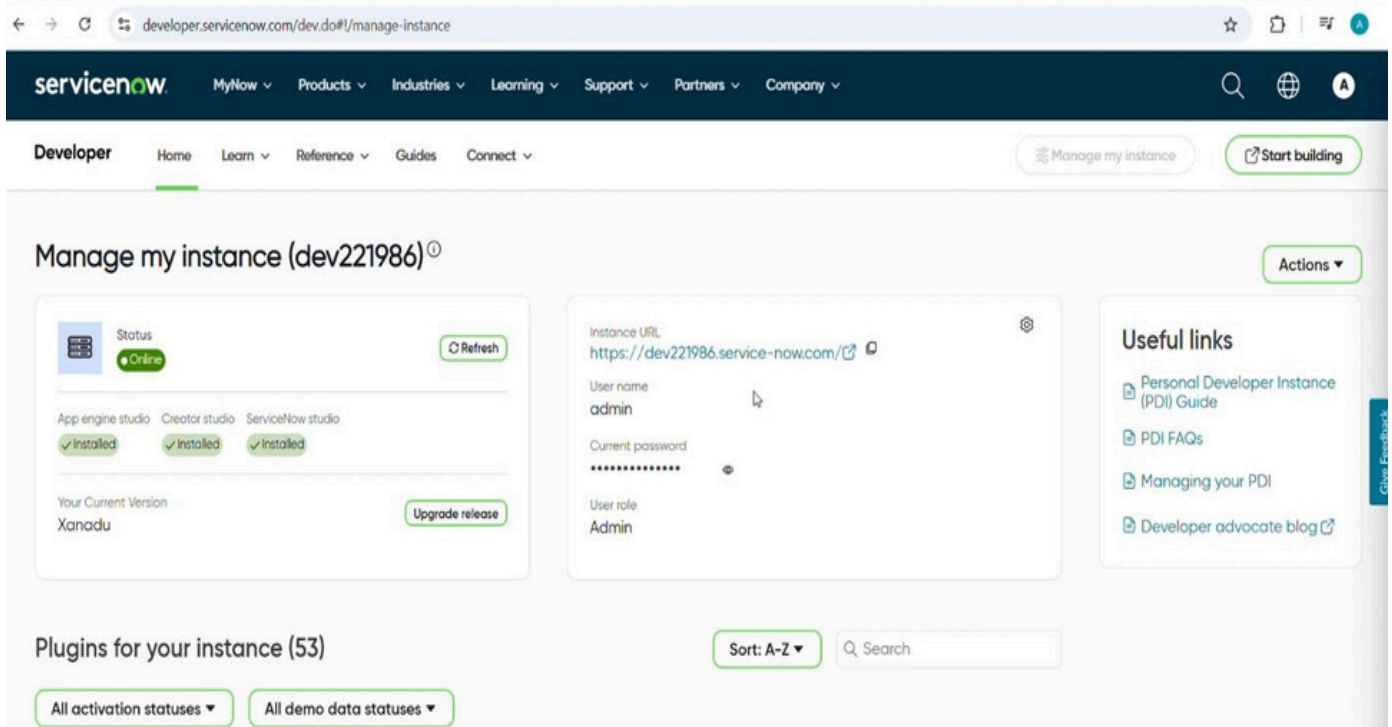
## KNOWLEDGE GAINED

- Learned how to set up and configure a ServiceNow developer instance for building applications. Understood the importance of update sets for tracking and migrating customizations.
- Gained practical skills in creating custom tables and defining fields to store structured data.
- Learned how to establish relationships between tables for linked data management.
- Practiced configuring related lists for easier navigation and record visibility.
- Understood how to create and apply business rules for automation and validations.
- Gained insights into data modeling and database concepts within ServiceNow.
- Learned how to generate reports and dashboards for real-time analysis.
- Understood how ServiceNow can be applied to non-IT use cases like family expense tracking.
- Improved overall knowledge of workflow automation and low-code development.



# MILESTONE 1: SETTING UP THE SERVICE NOW INSTANCE

- Go to the official ServiceNow Developer portal <https://developer.servicenow.com> and create a developer account.
- After signing in, open the Personal Developer Instance from the dashboard.
- Select Request Instance to generate a fresh environment for development.
- Provide the necessary details (like version selection) and
- Wait for the confirmation email containing your instance confirm your request.
- Use the credentials to log in to your newly created instance and login credentials.
- Once inside, explore the interface and begin working on the platform.



## MILSTONE 2: CREATION OF NEW UPDATE SET

- Log in to your ServiceNow instance and go to the Application Navigator.
- Search for Update Sets and open Local Update Sets under System Update Sets.
- Click on New to create a fresh update set.

set.

- o Name: Family Expenses
- Enter the following details:
  - o Description: Update set to capture all configurations related to the Family Expense Management project
- Save the record and mark it as the Current Update Set so every change you make is tracked under this set.
- Verify that the update set is active by checking the at the top of the screen.
- From this point forward, all customizations (relationships, and business rules) will be recorded inside the Family Expenses update set.

The screenshot shows the ServiceNow web interface for creating a new update set. The browser address bar displays the URL: dev221986.service-now.com/now/nav/ui/classic/params/target/sys\_update\_set.do%3Fsys\_id%3D-1%26sys\_is\_list%3Dtrue%26sys\_target%3Dsys\_update\_set%26sysparm\_checked\_items%3D... The page title is 'Update Set - Create New Update Set'. The breadcrumb trail shows 'Update Set' and 'New record'. The form contains the following fields: 'Name' (text input with value 'Family Expenses'), 'State' (dropdown menu with value 'In progress'), 'Parent' (text input with a search icon), 'Release date' (text input with a calendar icon), and 'Description' (large text area). The 'Application' field is set to 'Global'. At the bottom left, there are two buttons: 'Submit' and 'Submit and Make Current'. A mouse cursor is hovering over the 'Submit and Make Current' button.

# MILSTONE 3: CREATION OF TABLE FAMILY EXPENSES

## Activity 1 – Creating the Family Expenses Table

- In your ServiceNow instance, navigate to All > Tables using the filter navigator.
- Click on New to create a new table.
- Fill in the required details:
  - o Label: Family Expenses
  - o Name: (This will be auto-generated based on the label)
  - o New Menu Name: Family Expenditure
- Save the record to create the new table.

The screenshot shows the ServiceNow 'Table - New Record' form. The form is titled 'Table - New Record' and has a search bar. Below the title bar, there is a yellow banner with a message: 'ServiceNow recommends creating custom tables in scoped applications. To learn more about creating scoped applications, click [here](#).' Below this, a blue banner explains: 'A table is a collection of records in the database. Each record corresponds to a row in a table, and each field on a record corresponds to a column on that table. Applications use tables and records to manage data and processes. [More Info](#)'.

The form contains several fields and checkboxes:

- \* Label: Family Expenses
- \* Name: u\_family\_expenses
- Extends table: (empty field)
- Application: Global
- Create module: ☒
- Create mobile module: ☒
- Add module to menu: -- Create new --
- New menu name: Family Expenses

Below the form, there are tabs for 'Columns', 'Controls', and 'Application Access'. The 'Columns' tab is selected, showing a table with columns: Column label, Type, Reference, Max length, Default value, and Display. The table is currently empty, with a row for 'Insert a new row...'.

# MILSTONE 3: CREATION OF TABLE FAMILY EXPENSES

## Activity 2 – Adding Columns to the Family Expenses Table

Duration: 1 Hour Skill Tags: Table Configuration, Data Modeling, ServiceNow Basics

- Open the newly created Family Expenses table.
- To add columns, double-click near the existing columns to insert a new row.
- Enter the following details one by one:

1. Column Label: Number

2. Column Label:

6. Type: String

Date

o Type: Date

3. Column Label:

Amount

o Type: Integer

4. Column Label: Expense

Details

o Type: String

o Max Length: 800

Column label	Type	Reference	Max length	Default value	Display
Updates	Integer	(empty)	40	false	false
Updated	Date/Time	(empty)	40	false	false
Created by	String	(empty)	40	false	false
Sys ID	Sys ID (GUID)	(empty)	32	false	false
Updated by	String	(empty)	40	false	false
Created	Date/Time	(empty)	40	false	false
Number	String			false	false
Date	Date			false	false
Amount	Integer			false	false
Expense	String		800	false	false

# MILSTONE 3: CREATION OF TABLE FAMILY EXPENSES

Activity 3 –Making the Number Field an Auto-Number Open the Family Expenses table.

- Locatethe Number field/column and double-click to open itsproperties.
- Scroll down and switch to the Advanced
- view.
  - o Check the box for Use Dynamic Default.
  - o Set the Dynamic Default Value to Get Next Padded Number
- Click Update to save the changes.

The screenshot shows the ServiceNow interface for configuring a Dictionary Entry - Number. The page is titled "Dictionary Entry - Number" and is in the "Advanced" view. The "Default Value" tab is selected, showing the "Use dynamic default" checkbox checked and the "Dynamic default value" set to "Get Next Padded Number". The "Update" button is visible at the bottom.

dev221986.service-now.com/now/nav/ui/classic/params/target/sys\_dictionary.do%3Fsys\_id%3De8340a2583732210639fc396feaad3ba%26sysparm\_view%3Dadvanced

servicenow All Favorites History Workspaces Dictionary Entry - Number

Dictionary Entry Number View: Advanced

\* Max length 40 Mandatory Display

Attributes

Choice List Specification Calculated Value Default Value

The Default value specifies what value the field has when first displayed.

Use dynamic default ☒

Dynamic default value Get Next Padded Number

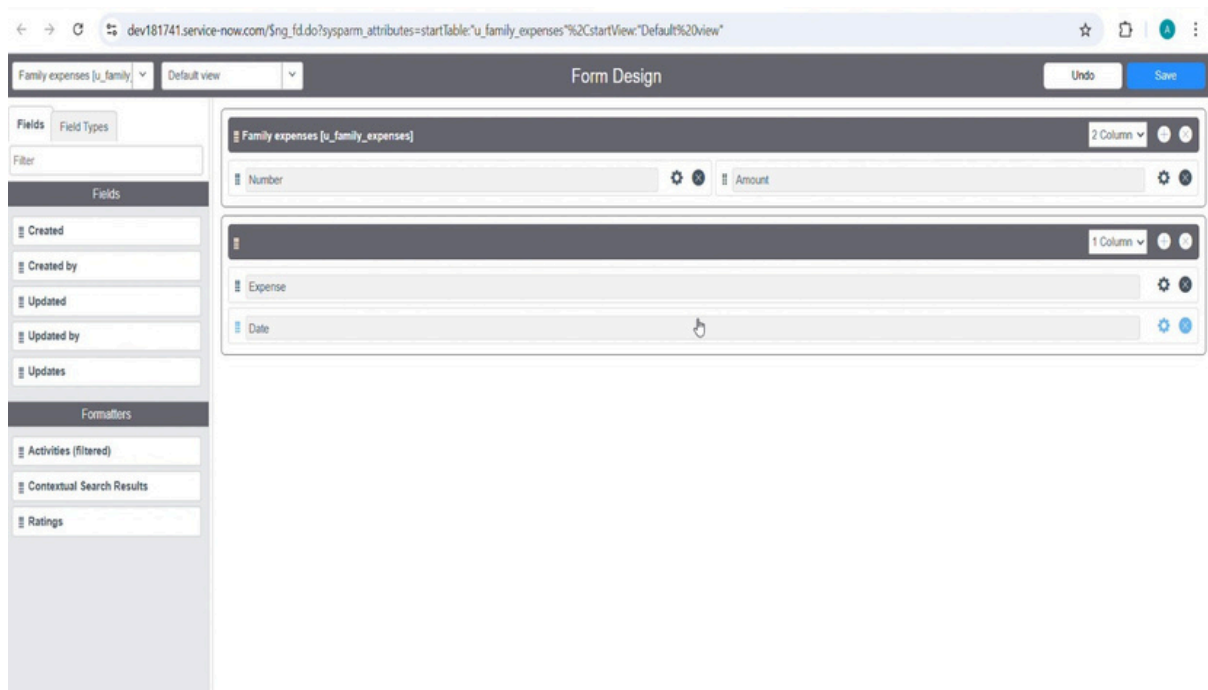
Delete Column Update

Related Links

# MILSTONE 3: CREATION OF TABLE FAMILY EXPENSES

## Activity 4 – Configuring the Form

- Navigate to All > in the filter, search for Family
- Expenses. Open the Family Expenses table. Click on
- New to create a new form entry. On the form header,
- right-click and select:
  - o Configure > Form Design.
- In the Form Designer, use drag-and-drop to:
  - o Rearrange fields.
  - o Group related fields together.
  - o Add sections if required for better clarity.
- Save the customized form layout.



# MILSTONE 4: CREATION OF TABLE DAILY EXPENSES

## Activity 1 – Creating The Daily Expenses Table

- Navigate to All > Tables using the filter
- navigator. Click on New to create a new table.
- Fill in the required details:
- Label: Daily Expenses
- Name: (Auto-populated by the system)
- Add Module to Menu: Family Expenditure
- Go to the form header, right-click, and select Save.

dev181741.service-now.com/now/nav/ui/classic/params/target/sys\_db\_object.do%3Fsys\_id%3D-1%26sys\_is\_list%3Dtrue%26sys\_target%3Dsys\_db\_object%26sysparm\_checked\_items%3D%26sys...

servicenow All Favorites History Workspaces Admin Table - New Record Search Submit Cancel

ServiceNow recommends creating custom tables in scoped applications. To learn more about creating scoped applications, click [here](#).

A table is a collection of records in the database. Each record corresponds to a row in a table, and each field on a record corresponds to a column on that table. Applications use tables and records to manage data and processes. [More Info](#)

\* Label: Daily Expenses Application: Global

\* Name: u\_daily\_expenses Create module: ☒

Extends table: Create mobile module: ☒

Add module to menu: -- Create new -- New menu name: Daily Expenses

Remote Table: ☐

Columns Controls Application Access

Table Columns for text Search

Dictionary Entries

Column label	Type	Reference	Max length	Default value	Display
--------------	------	-----------	------------	---------------	---------

# MILSTONE 4: CREATION OF TABLE DAILY EXPENSES

## Activity 2 – Creating Columns (Fields)

- Open the Daily Expenses
- **Table Columns**, double-click to insert a new row and add the fields:

1. Column Label:

- Type: Number
- String

2. Column Label:

- Type: Date

3. Column Label:

- Type: Expense
- Integer

4. Column Label: Family Member

- Type: Reference
- Max Length: 800

5. Column Label:

- Comments
- Type: String
- Max Length: 800

Table - Daily Expenses				
Updated by	String	(empty)	40	false
Updates	Integer	(empty)	40	false
Updated	Date/Time	(empty)	40	false
Created by	String	(empty)	40	false
Created	Date/Time	(empty)	40	false
Sys ID	Sys ID (GUID)	(empty)	32	false
Number	String			false
Date	Date			false
Expense	Integer			false
Family member	Reference			false

Insert a new row...

Delete Update Delete All Records

Related Links

- Form Builder
- Design Form
- Layout Form
- Layout List
- Show Form



# MILSTONE 4: CREATION OF TABLE DAILY EXPENSES

## Activity 3 – Making Number Field an Auto-Number

- Open the Daily Expenses
- ~~Locate~~ <sup>Table</sup> the Number field/column and double-click to open its properties
- Scroll down and switch to the Advanced View. In the Default Value section:
  - o Check the box for Use Dynamic Default.
  - o Set the Dynamic Default Value to Get Next Padded Number.
- Click Update to save

## Configuring Number

Main Navigation: to All > Number Maintenance.

- Click on New. Enter the
- details as follows:
  - o Table: Family Expenses
  - o Prefix: MFE
- Click on Submit.

The screenshot shows the ServiceNow interface for configuring a 'Dictionary Entry - Number'. The browser address bar shows a URL with a long alphanumeric string. The page title is 'Dictionary Entry - Number' with a star icon. Below the title bar, there are tabs for 'Choice List Specification', 'Calculated Value', and 'Default Value', with 'Default Value' being the active tab. A blue informational box states: 'The Default value specifies what value the field has when first displayed.' Below this, there is a checkbox labeled 'Use dynamic default' which is checked. Underneath, there is a text field labeled 'Dynamic default value' containing the text 'Get Next Padded Number'. To the right of this field are search and help icons. At the bottom left of the configuration area are 'Delete Column' and 'Update' buttons. Below the configuration area is a 'Related Links' section with links for 'Show Table', 'Run Point Scan', and 'Default view'. At the very bottom, there is a table with columns: 'Name', 'Decision Type', 'Operation', 'Type', 'Active', 'Updated by', and 'Updated'. The table is currently empty.

dev181741.service-now.com/now/nav/ui/classic/params/target/sys\_number.do%3Fsys\_id%3D-1%26sys\_is\_list%3Dtrue%26sys\_target%3Dsys\_number%26sysparm\_checked\_items%3D%26syspar...

servicenow

AllFavoritesHistoryWorkspacesAdmin

Number - Create DFE ☆

Search

Number

New record

Submit

⊗ Invalid insert ×

\* Table

Daily Expenses

Prefix

DFE

\* Number

1,000

Application

Global

Number of digits

7

Submit

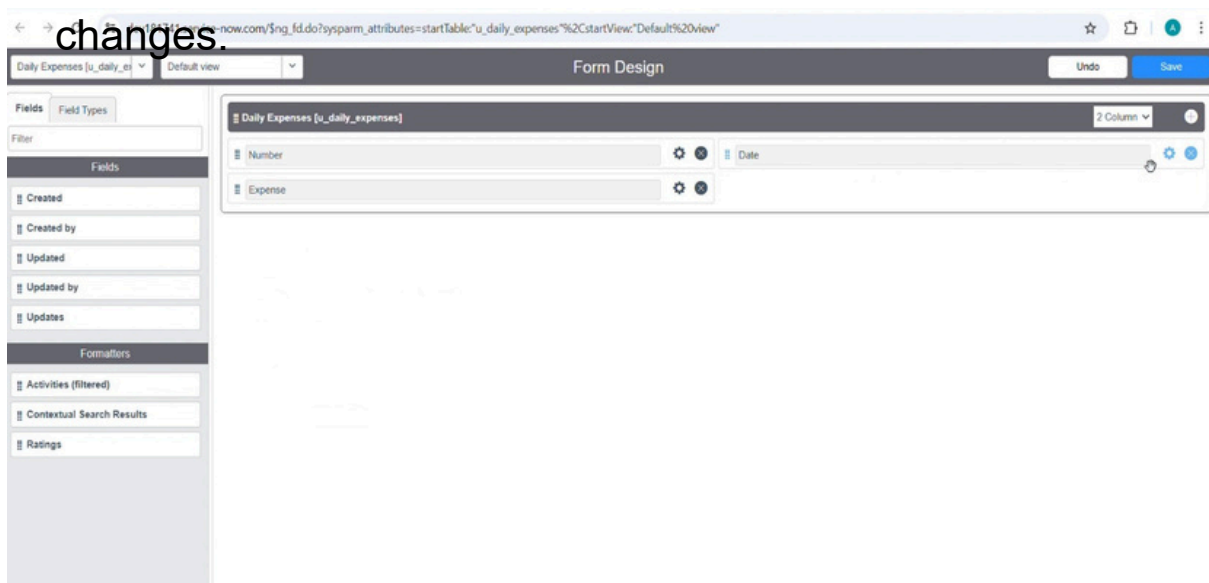
Related Links

[Show Counter](#)

# MILSTONE 4: CREATION OF TABLE DAILY EXPENSES

## Activity 4 – Configure The Form

- Navigate to All > Daily Expenses using the filter. Open the Daily Expenses table. Click on New to create a new form entry. On the form header, right-click, then select:
- Configure > Form Design.
- In the Form Designer, drag and drop fields to customize the form layout as per requirement. Apply the following configurations:
- Number Field → Set as Read-Only by clicking the gear icon and checking Read-Only.
- Date Field → Set as Mandatory by clicking the gear icon and checking Mandatory.
- Family Member Name Field → Set as Mandatory using the same method.
- Click Save to apply the changes.



# MILSTONE 5: CREATION OF RELATIONSHIP BETWEEN FAMILY EXPENSES AND DAILY EXPENSES TABLES

- Navigate to All > Relationships using the filter navigator. New to create a new relationship. Fill in the details as follows:
  - Name: Daily Expenses
  - Applies to Table: Family Expenses
  - Related List Table: Daily Expenses
- Click Save.

dev181741.service-now.com/now/nav/ui/classic/params/target/sys\_relationship.do%3Fsys\_id%3D834e116c3fb6210555b3942b4013151%26sysparm\_view%3D%26sysparm\_domain%3Dnull%26...

servicenow All Favorites History Workspaces Relationship - Daily Expenses Search

Relationship Daily Expenses Update Delete

Name:  Application:

Advanced ☐ Applies to table:  Queries from table:

This script refines the query in current that will populate the related list. For more information about it, its parameters and control variables, see [the documentation](#) See also the article about the [recommended form of the script](#).

Query with ☐ Turn on ECMAScript 2021 (ES12) mode

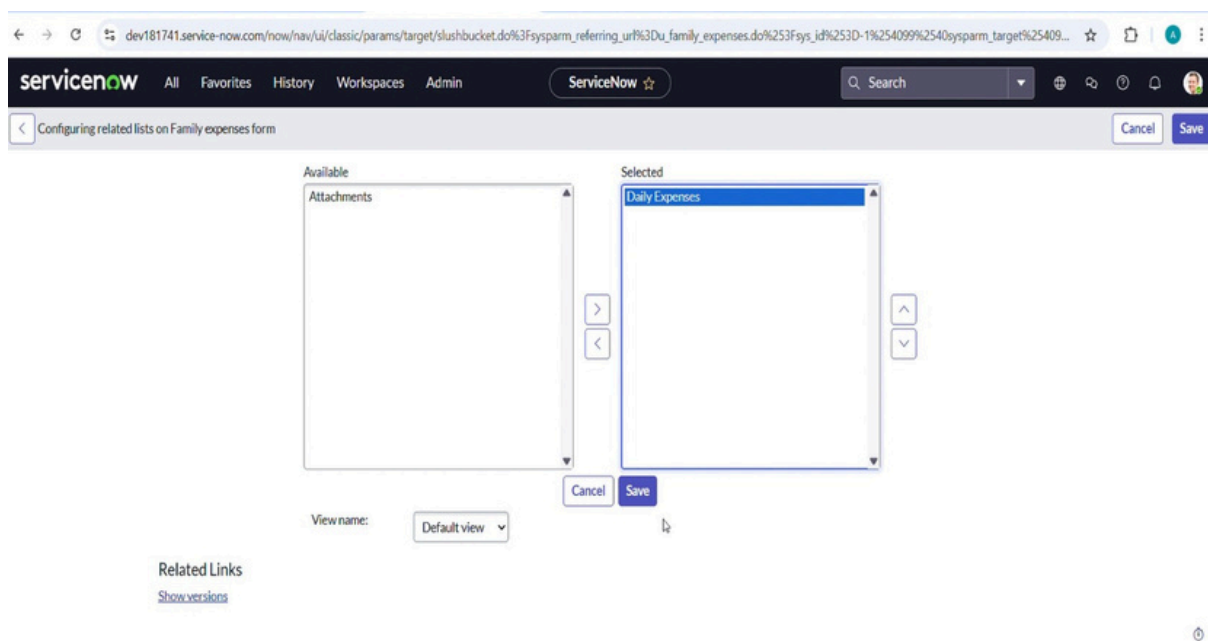
```
1 (function refineQuery(current, parent) {
2
3     // Add your code here, such as current.addQuery(field, value);
4
5 })(current, parent);
```

Update Delete

Related Links  
[Run Point Scan](#)

## MILSTONE 6: CONFIGURING RELATED LIST ON FAMILY EXPENSES

- Navigate to All > Family Expenses using the filter.
- Open the Family Expenses table. Click on New to open the form view. On the form header, right-click, then select:
  - Configure > Related Lists.
  - From the available options, add Daily Expenses to the Selected Area. Click Save to apply the changes.
- 



## MILSTONE 7: CREATION OF BUSINESS RULES

- NavigatetoAll >Business Rules usingthe filter. Under
- System Definition, select Business Rules and click New.
- Enter the following details:
  - o Name: Family Expenses
  - BR o Table: Daily Expenses
  - o Check Advanced.
- In the When to run section,  
checkInsert
  - o Update
- In the Advanced tab, add the following script: (function  
executeRule(current, previous /\*null when async\*/) {

```
varFamilyExpenses = new
GlideRecord('u_family_expenses');
FamilyExpenses.addQuery('u_date', current.u_date);
FamilyExpenses.query();
if(FamilyExpenses.next()) {
    FamilyExpenses.u_amount += current.u_expense;
    FamilyExpenses.u_expense_details += ">" +
":"+"Rs." + current.u_expense + "/-";
    FamilyExpenses.update
}else {
    ();
    var NewFamilyExpenses = new

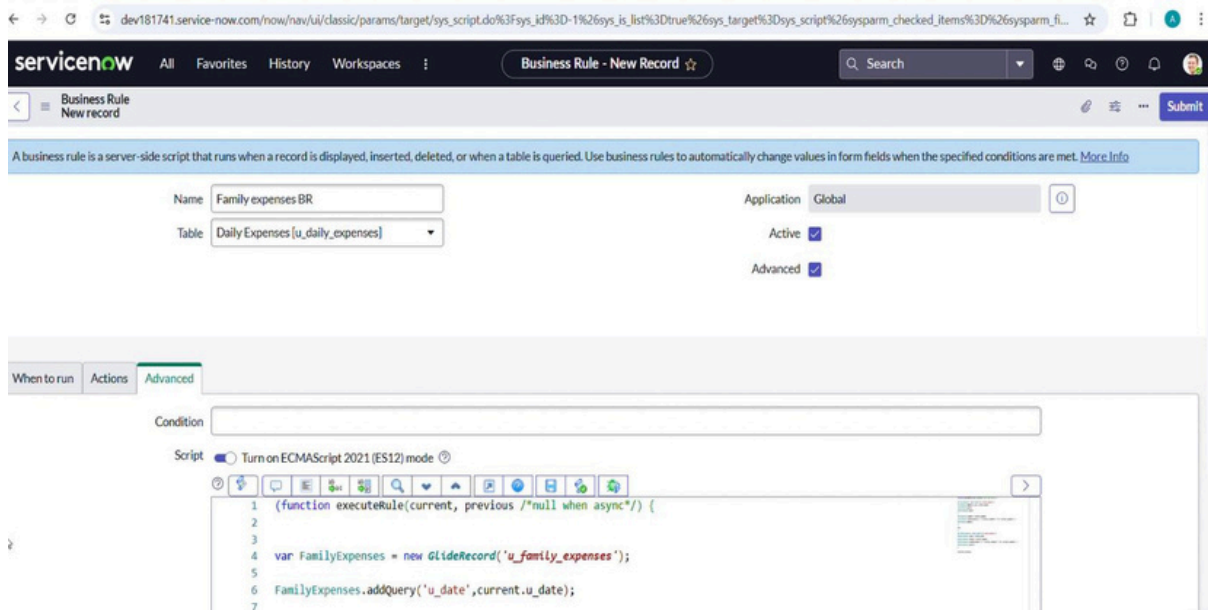
    GlideRecord('u_family_expenses');

    NewFamilyExpenses.u_date = current.u_date;

    NewFamilyExpenses.u_amount = current.u_expense;
current.u_comments + ":" + "Rs." + current.u_expense +
"/-"; NewFamilyExpenses.u_expense_details += ">" +
```

```
NewFamilyExpenses.insert(
    } );
})(current,
previous);
```

Go to the form header, right-click, then select  
Save.



## MILSTONE 8: CONFIGURE THE RELATIONSHIP

- Navigate to All > Relationships using the filter
- navigator. Open the existing Daily Expenses
- Relationship. Update the details as follows:
  - o Applies to Table: Family
- In **Refine Query** with section, enter the following script: (function refineQuery(current, parent) {

```
// Add your code here, such as current.addQuery(field, value); current.addQuery('u_date', parent.u_date); current.query(); })(current, parent);
```

Click **Update** to save the

configuration

servicenow All Favorites History Workspaces Relationship - Daily Expenses Search

Relationship Daily Expenses Update Delete

Name: Daily Expenses Application: Global

Advanced Applies to table: Family expenses [u\_family\_expenses] Queries from table: Daily Expenses [u\_daily\_expenses]

This script refines the query in current that will populate the related list. For more information about it, its parameters and control variables, see [the documentation](#). See also the article about the [recommended form of the script](#).

Query with Turn on ECMAScript 2021 (ES12) mode

```
4 // Add your code here, such as current.addQuery(field, value);
5
6 current.addQuery('u_date', parent.u_date);
7
8 current.query();
9
10
11 })(current, parent);
```

Update Delete

Related Links



## CONCLUSION

The Family Expenses Management System built on ServiceNow demonstrates how the platform can be leveraged beyond IT workflows to solve real-world problems. By systematically creating tables, relationships, forms, and business rules, the project enables seamless tracking of both daily expenses and family-level expenses in an automated manner.

The use of auto-numbering, mandatory fields, related lists, and business rules ensures data integrity, consistency, and accuracy. The relationship configuration further enhances visibility by linking daily records to family-level summaries, providing a clear financial overview.

Through this project, we learned how to apply ServiceNow features such as table creation, form design, field configuration, scripting, and automation to build a complete application. More importantly, it highlights how low-code/no-code platforms like ServiceNow can be extended into personal finance, household management, and non-IT use cases.

Overall, the project provides a practical, scalable, and user-friendly solution for managing family expenses efficiently while also strengthening skills in ServiceNow application development.