

Calculating Family Expenses using Service Now

Team ID: NM2025TMID07649

Team Size: 5

Team Leader: Praveen Kumar S

Team member: Ravi P

Team member: Raja Kabin R K

Team member: Rathi A

Team member: Raja Lakshmi A

AIM:

To perform calculating familiy expenses using ServiceNow Developer.

PROBLEM STATEMENT:

- Managing household finances is essential for every family to ensure financial stability and better planning. This project aims to develop a simple and efficient system using ServiceNow to record, track, and calculate monthly family expenses. ServiceNow, being a powerful cloud-based platform, offers custom application development that can be leveraged for personal finance tracking.

OBJECTVE:

- To build a Family Expense Management System using ServiceNow that allows users to:
- Enter daily or monthly expenses under different categories (e.g., groceries, rent, transport, education, etc.)
- View summary reports of total and category-wise expenses
- Set monthly budgets and receive alerts if spending exceeds limits
- Generate downloadable reports for any time period

SKILL:

1. ServiceNow Application Development

- Creating custom tables
- Designing forms and views
- Using Flow Designer for automation

2. Workflow & Automation

- Using ServiceNow Flow Designer to automate calculations and notifications

3. Database Management

- Creating and managing custom tables for storing expenses and budgets

4. Reporting & Dashboard Design

- Designing ServiceNow reports

Building graphical dashboards for data visualization

TASK INITIATION:

STEP 1: Setting up ServiceNow Instance

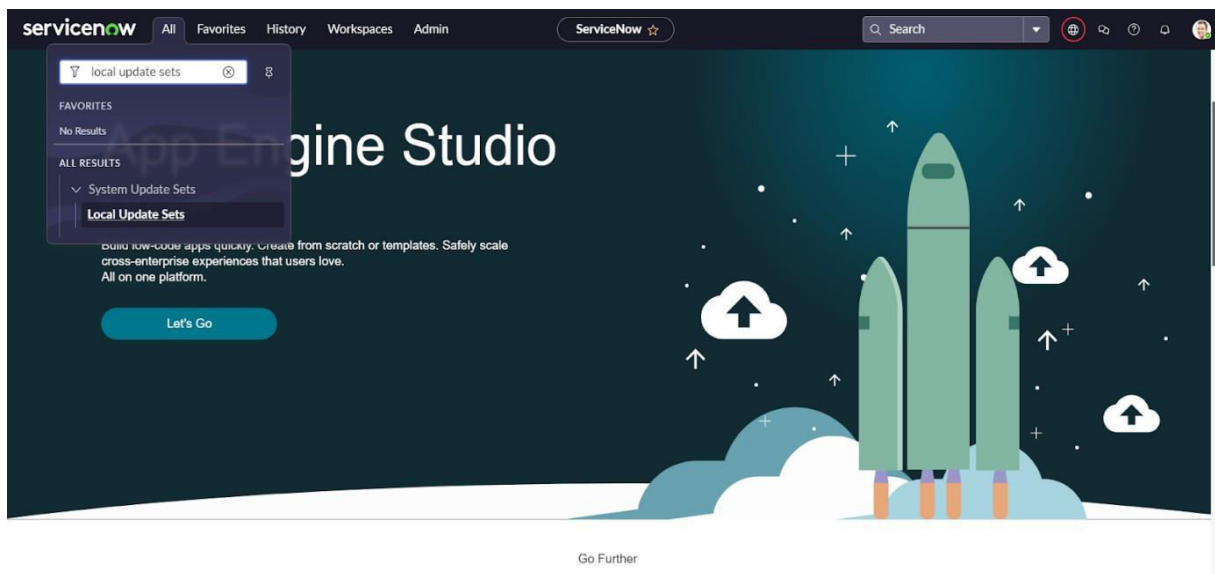
ACTIVITY 1: Setting up ServiceNow Instance

1. Sign up for a developer account on the ServiceNow Developer site
“<https://developer.servicenow.com>”.
2. Once logged in, navigate to the "Personal Developer Instance" section.
3. Click on "Request Instance" to create a new ServiceNow instance.
4. Fill out the required information and submit the request.
5. You'll receive an email with the instance details once it's ready.
6. Log in to your ServiceNow instance using the provided credentials.
7. Now you will navigate to the ServiceNow.

STEP 2: Creation of New Update Set

ACTIVITY 1: Creation of New Update Set

1. Go to All >> In the filter search for Local Update set > click on New.



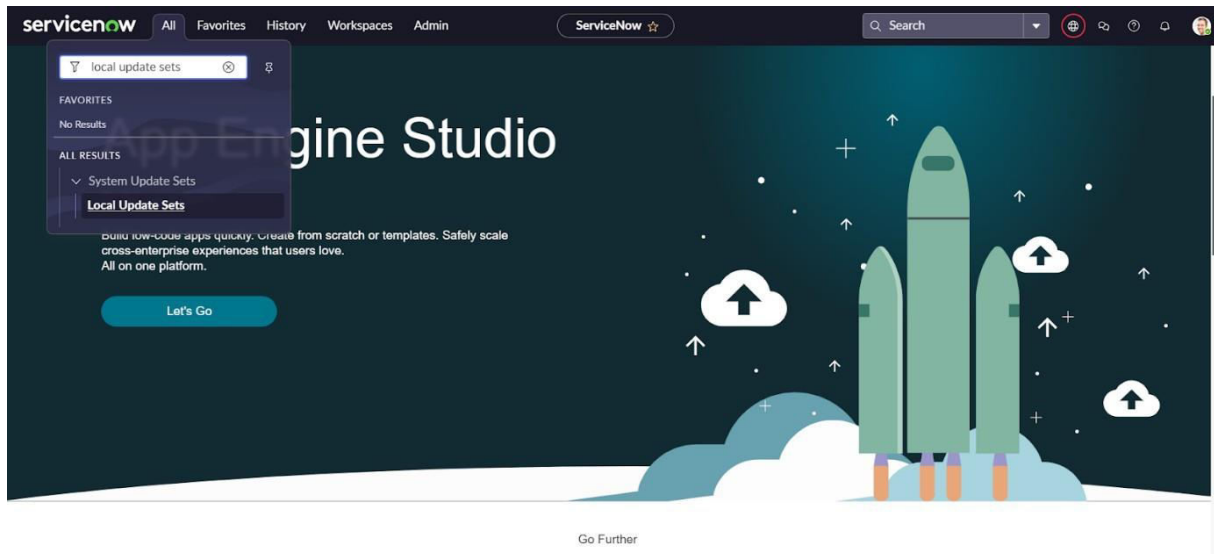
2. Enter the Details as: Name:

Family Expenses

3. Then click on Submit and Make current.

STEP 3: Creation of Table ACTIVITY 1: Creation of New Update Set

1. Go to All >> In the filter search for Local Update set > click on New.



1. Enter the Details as: Name: Family Expenses
2. Then click on Submit and Make current

The screenshot shows the 'Update Set - Create Default 2' form in ServiceNow. The form has a header bar with 'Update Set - Create Default 2' and a search bar. Below the header, there's a message bar stating 'You can now publish customizations to the Application Repository. Go to the Application record and use Related Links.' The form fields are: Name (Family Expenses), State (In progress), Parent (empty), Release date (empty), and Description (empty). The Application is set to ServiceNow IDE. At the bottom, there are buttons for 'Submit' and 'Submit and Make Current'.

STEP 4: Creation of Table

ACTIVITY 1: Creation of Family Expenses Table

1. Go to All > In the filter search for Tables > click on New.
2. Enter the Details:

Label: Family Expenses Name: Auto-Populated

New menu name: Family Expenditure

The screenshot shows the ServiceNow 'Table - New Record' form. The top navigation bar includes 'servicenow', 'All', 'Favorites', 'History', 'Workspaces', 'Admin', and a 'Table - New Record' button. A search bar is also present. The form itself has a header with a back arrow, a hamburger menu, and the text 'Table New record'. Below this is a blue informational banner. The main form area contains several fields: 'Label' with the value 'Family Expenses', 'Name' with the value 'sn_glider_family_expenses', 'Application' with the value 'ServiceNow IDE', and 'Extends table' which is empty. There are also checkboxes for 'Create module' and 'Create mobile module', both of which are checked. At the bottom right, there is a dropdown menu for 'Add module to menu' with the option 'Create new...'. The form concludes with 'Submit' and 'Cancel' buttons.

1. Go to the Header and right click there>> click on Save.
2. ACTIVITY 2: Creation of Columns (Fields)

1. Near Columns Double click near insert a new row.

2. Give the details as: Column

label: Number Type: String

3. Double clicks on insert a new row again

4. Give the details as:

Column label: Date

Type: Date

5. Double clicks on insert a new row again

6. Give the details as: Column

label: Amount Type: Integer

7. Double clicks on insert a new row again

8. Give the details as:

Column label: Expense Details Type: String

Max length: 800

Columns Controls Application Access						
Table Columns for text Search 1 to 11 of 11 New						
Dictionary Entries						
	Column label	Type	Reference	Max length	Default value	Display
	Updates	Integer	(empty)	40		false
X	Comments	String	(empty)	800		false
	Updated by	String	(empty)	40		false
X	Expense	Integer	(empty)	40		false
X	Number	String	(empty)	40	javascript:getNextObjNumberPadded();	false
	Created by	String	(empty)	40		false
X	Date	Date	(empty)	40		false
	Sys ID	Sys ID (GUID)	(empty)	32		false
	Updated	Date/Time	(empty)	40		false
	Created	Date/Time	(empty)	40		false

1. Go to the Header and right click there>> click on Save.

ACTIVITY 3: Making Number Field an Auto- Number

1.Double click on the Number Field/Column

2.Go down and double click on Advanced view

3.In Default Value:

Use dynamic default: check the box Dynamic default

value: Get Next Padded

Number

1. Click on Update.

The screenshot shows the SAP Field Catalog configuration for a Number field. The configuration is divided into several sections:

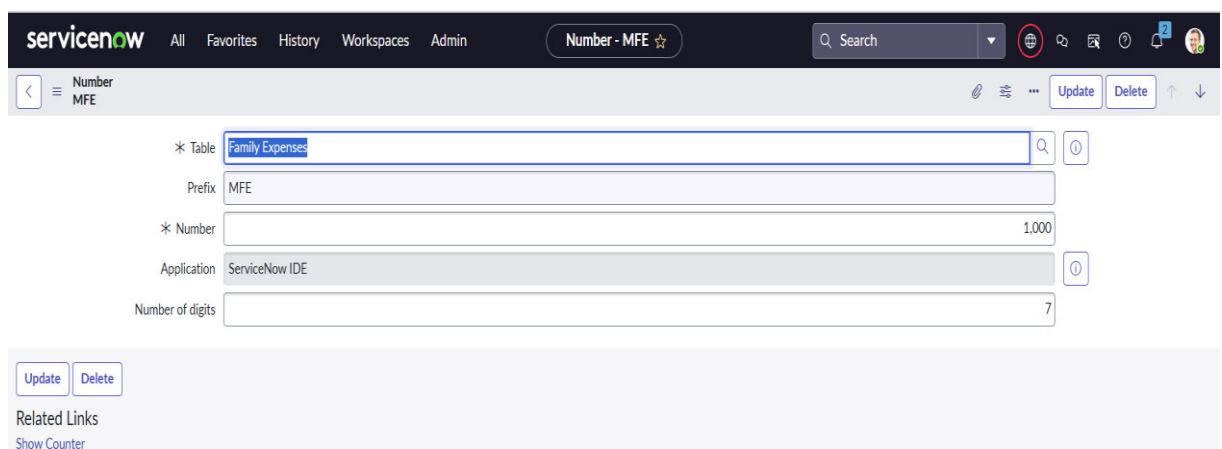
- Basic Data:** Includes fields for Type (String), Column label (Number), Column name (u_number), and Max length (40).
- Advanced View:** Includes checkboxes for Active, Function field, Read only, Mandatory, and Display.
- Attributes:** A section for altering the behavior of a field or functionality that depends on the field, with a link to [More Info](#).
- Default Value:** A section for specifying the default value. It includes a checkbox for "Use dynamic default" (checked) and a text field for "Dynamic default value" (Get Next Padded Number).
- Buttons:** "Delete Column" and "Update" buttons are located at the bottom.
- Related Links:** A section for related links.

1. Go to All >> In the filter search for Number Maintenance>>select Number Maintenance

2.Click on New

3.Enter the below Details: Table:

Family Expenses Prefix: MFE



The screenshot shows the ServiceNow interface for configuring a Number. The top navigation bar includes 'servicenow', 'All', 'Favorites', 'History', 'Workspaces', 'Admin', and a search bar. Below the navigation bar, the breadcrumb trail shows 'Number' > 'MFE'. The main form contains the following fields:

- * Table: (with a search icon and a help icon)
- Prefix:
- * Number:
- Application: (with a help icon)
- Number of digits:

At the bottom of the form, there are 'Update' and 'Delete' buttons. Below the buttons, there is a 'Related Links' section with a link to 'Show Counter'.

9.Click on Submit.

ACTIVITY 4: Configure the Form

1. Go to All >> In the filter search for Family Expenses
>> Open Family Expenses

2. Click on New

3. Go to the Header and right click there>> click on Configure
>> Select Form Design

4. Customize or Drag Drop the form as per your requirement.

Family Expenses [u_family_expenses] 2 Column

Number Date Amount

Expense Details 1 Column

5. Make Number Read-Only Field by clicking on the gear icon and checking Read-Only

6. Make Date, Amount Mandatory Field by clicking on the gear icon and checking Mandatory

7. Click on Save.

STEP 5: Creation of Table (Daily Expenses)

ACTIVITY 1: Creation of Daily Expenses Table

1. Go to All > In the filter search for Tables > click on New.

2. Enter the Details:

Label: Daily Expenses Name: Auto-Populated

Add Module to menu: Family Expenditure

serviceNow All Favorites History Workspaces Admin Table - New Record

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

* Name: sn_glider_daily_expenses

Extends table

Application: ServiceNow IDE

Create module ☒

Create mobile module ☒

Add module to menu: -- Create new --

New menu name: Family Expenditure

Remote Table ☐

Columns Controls Application Access

1. Double clicks on insert a new row again
2. Give the details as: Column
label: Expense Type: Integer
3. Double click on insert a new row again
4. Give the details as:
Column label: Family Member Name Type:
Reference
Max length: 800
5. Double clicks on insert a new row again
6. Give the details as: Column
label: Comments Type: String
Max length: 800
7. Go to the Header and right click there>> click on
Save.

ACTIVITY 3: Making Number Field an Auto- Number

1. Double clicks on the Number Field/Column.
2. Go down and double click on Advanced view
3. In Default Value:

Use dynamic default: check the box

Dynamic default value: Get Next Padded Number
4. Click on Update.

* Type

String

🔍

🗑️

* Column label

Number

* Column name

u_number

* Max length

40

Active

☒

Function field

☐

Read only

☒

Mandatory

☐

Display

☐

Alters the behavior of a field or functionality that depends on the field. [More Info](#)

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

5. Go to All >> In the filter search for Number Maintenance >> select Number Maintenance
6. Click on New.
7. Enter the below Details: Table:
Family Expenses Prefix: MFE

servicenow

AllFavoritesHistoryWorkspacesAdmin

Number - DFE

Search

Number DFE

UpdateDelete

* Table

Daily Expenses

Q

?

Prefix

DFE

* Number

1,000

Application

ServiceNow IDE

?

Number of digits

7

Update

Delete

Related Links

9. Click on Submit.

ACTIVITY 4: Configure the Form

1. Go to All >> In the filter search for Daily Expenses
>> Open Daily Expenses
2. Click on New
3. Go to the Header and right click there>> click on
Configure >> Select Form Design
4. Customize or Drag Drop the form as per your
requirement.

The screenshot shows a form configuration window for 'Daily Expenses [u_daily_expenses]'. The form is currently set to a 2-column layout. It contains four fields: 'Number', 'Date', 'Family Member Name', and 'Expense'. Each field has a gear icon for configuration. Below this, there is a section for 'Comments' with a 1-column layout.

5. Make Number Read-Only Field by clicking on the gear icon and checking Read-Only

5. Make Date, Family Member Name Mandatory Field by clicking on the gear icon and checking Mandatory

6. Click on Save.

STEP 6: Creation of Relationship

ACTIVITY 1: Creation of Relationship between Family Expenses and Daily Expenses table

1. Go to All >> In the filter search for Relationships >> Open Relationships

2. Click on New.

3. Enter the details:

Name : Daily Expenses

Applies to table : Select Family Expenses

Daily Expenses : Select Daily Expenses

4. Click Save.

STEP 7: Configuring Related List on Family Expenses

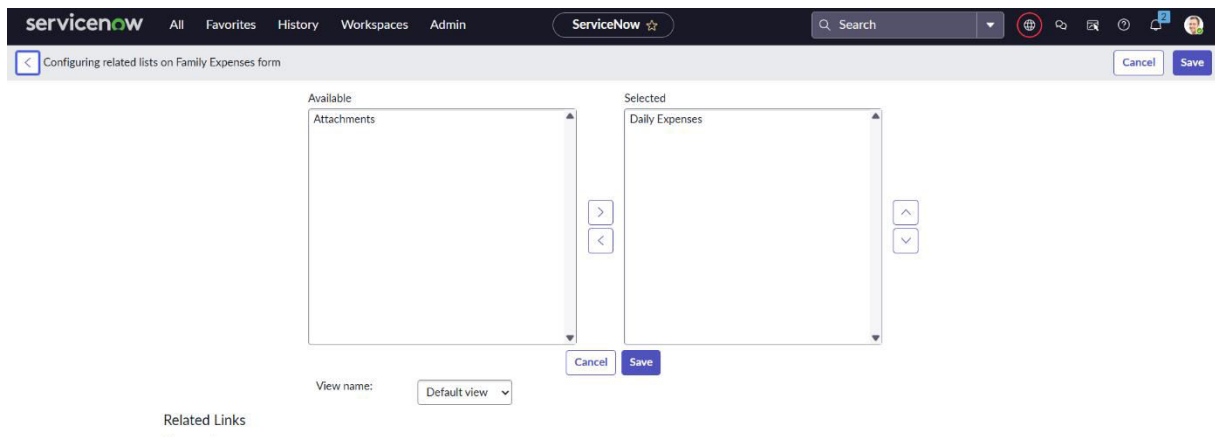
1. Go to All >> In the filter search for Family Expenses >> Open Family Expenses

2. Click on New

3. Go to the Header and right click there>> click on Configure >> Select Related Lists

4. Add Daily Expenses to the Selected Area.

5. Click on Save

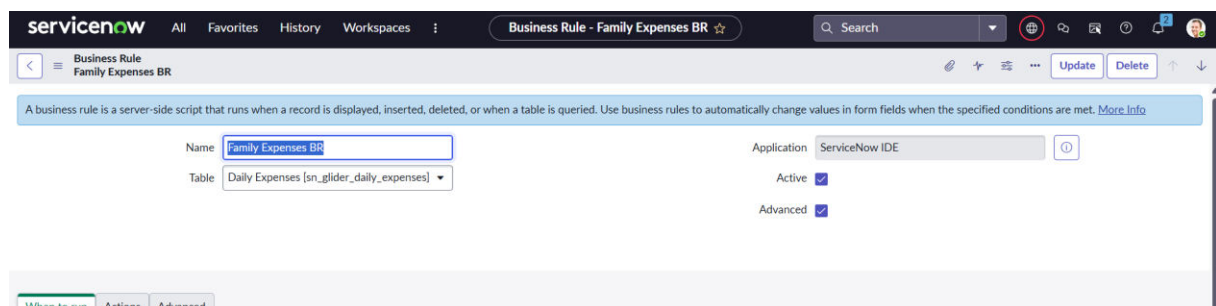


STEP 8: Creation of Business Rules

1. Go to All >> In the filter search for Business Rules.
2. Under System Definition Select Business Rules then click on New.
3. Enter the Details:

Name : Family Expenses BR

Table : Select Daily Expenses Check Advanced



4. In when to run Check Insert and Update

servicenow All Favorites History Workspaces : Business Rule - Family Expenses BR

Business Rule Family Expenses BR

When to run Actions Advanced

Specify whether the business rule should run on **Insert** or **Update**. Use **Filter Conditions** to specify under which conditions the business rule should run.

When before

Order 100

Insert ☒

Update ☒

Delete ☐

Query ☐

Filter Conditions Add Filter Condition Add OR Clause

-- choose field -- -- oper -- -- value --

Role conditions

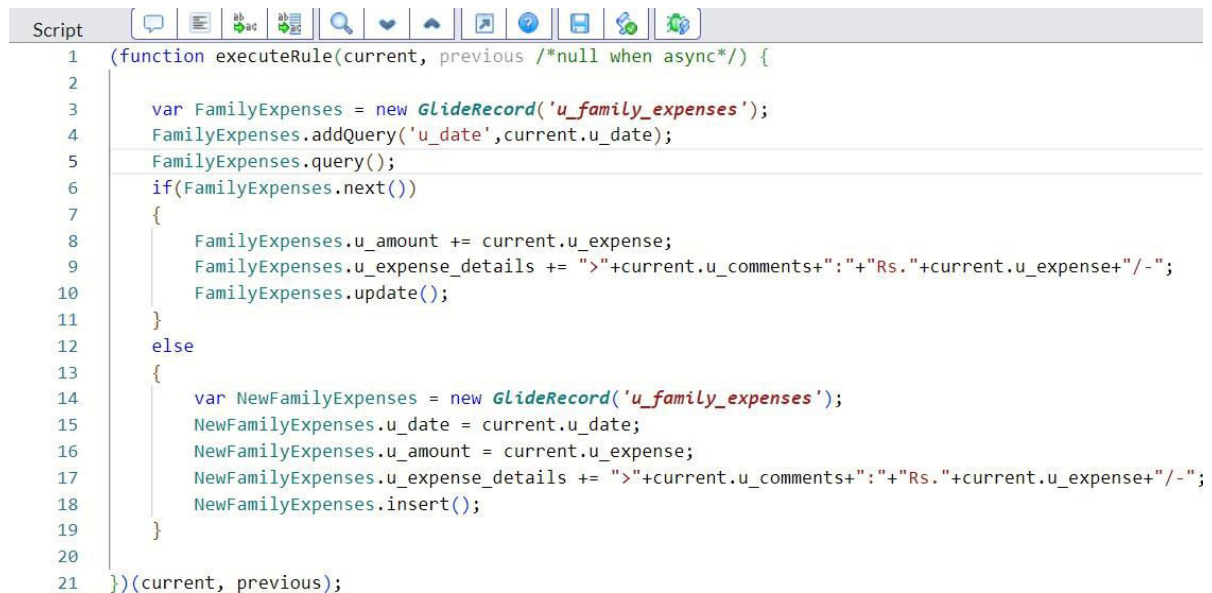
1. In Advance (we write the code): Write the below code

```
(function executeRule(current, previous /*null when async*/) {
    var FamilyExpenses = 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 +=
        ">" + current.u_comments + ":" + "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;
        NewFamilyExpenses.u_expense_details +=
        ">" + current.u_comments + ":" + "Rs." + current.u_expense + "/-";
        NewFamilyExpenses.insert();
    }

})(current, previous);
```



```

1 (function executeRule(current, previous /*null when async*/) {
2
3     var FamilyExpenses = new GlideRecord('u_family_expenses');
4     FamilyExpenses.addQuery('u_date',current.u_date);
5     FamilyExpenses.query();
6     if(FamilyExpenses.next())
7     {
8         FamilyExpenses.u_amount += current.u_expense;
9         FamilyExpenses.u_expense_details += ">" + current.u_comments + "Rs." + current.u_expense + "/-";
10        FamilyExpenses.update();
11    }
12    else
13    {
14        var NewFamilyExpenses = new GlideRecord('u_family_expenses');
15        NewFamilyExpenses.u_date = current.u_date;
16        NewFamilyExpenses.u_amount = current.u_expense;
17        NewFamilyExpenses.u_expense_details += ">" + current.u_comments + "Rs." + current.u_expense + "/-";
18        NewFamilyExpenses.insert();
19    }
20
21 }(current, previous);

```

STEP 9: Configure the Relationship

1. Go to All >> In the filter search for Relationships
>> Open Relationships.
2. In that, open Daily Expenses Relationship.
3. For Applies to table : Select Family Expenses.
4. In Query with : write the below Query. (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);

servicenow All Favorites History Workspaces : Relationship - Daily Expenses ☆ Search

Relationship Daily Expenses Run Query Diagnostics Update Delete

Name Application ServiceNow IDE

Advanced ☐ Applies to table Family Expenses [sn_glider_family_expen...]

Simple reference ☐ Queries from table Daily Expenses [sn_glider_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

```
1 (function refineQuery(current, parent) {
2
3   // Add your code here, such as current.addQuery(field, value);
4
5   current.addQuery('u_date', parent.u_date);
6
7   current.query();
8
9 })(current, parent);
```

Run Query Diagnostics Update Delete

5. Click on Update.

Example Output:01

Example Output:02

servicenow

AllFavoritesHistoryWorkspacesAdmin

Family Expenses - 2

Search

2

Family Expenses2

UpdateDelete

Number

2

Date

2025-10-31

Expense Details

Mobile Recharge

Amount

350

UpdateDelete

Daily Expenses

Number

Search

New

Daily Expenses

Number

Comments

Date

Expense

Family Member Name

No records to display

Conclusion:

The Family Expense Calculation System on ServiceNow offers an easy and efficient way to manage household finances. With features like expense tracking, budget setting, and reporting, it helps users monitor spending and make informed financial decisions. The system's scalability and user-friendly interface ensure it supports families of all sizes, promoting better financial planning and control.