

Calculating Family Expenses using ServiceNow

Team ID: LTVIP2026TMIDS70988

Team Size: 4

Team Leader: Bokka Naveen Kumar Bokka

Team Member: Jagadeeswari Peddireddy

Team Member: Hema Nandini Ganti

Team Member: Gudala Ratnaraju

Problem Statement:

There is currently no efficient system in place to track, categorize, and analyze family expenses in a way that is integrated, automated, and accessible in real-time. This leads to poor budget planning, missed payments, overspending, and a lack of financial visibility.

Objective:

To design and implement a solution using ServiceNow that allows family members to:

1. **Input and categorize expenses** easily via a user-friendly interface or mobile portal.
2. **Automate recurring expenses** (e.g., rent, subscriptions).
3. **Track total and category-wise spending** on a monthly/weekly basis.
4. **Set spending limits and receive alerts** when thresholds are crossed.
5. **Generate reports and dashboards** to visualize trends and support better financial decisions.
6. **Share visibility and responsibilities** across family members via roles or user groups.

Skills:

- **ServiceNow App Development** (App Engine Studio)
- **Custom Tables & Data Modeling**
- **Form Design & UI Customization**
- **Flow Designer** (for automation)
- **Business Rules & Scripting**
- **Reporting & Dashboards**
- **User Roles & Access Control (ACLs)**

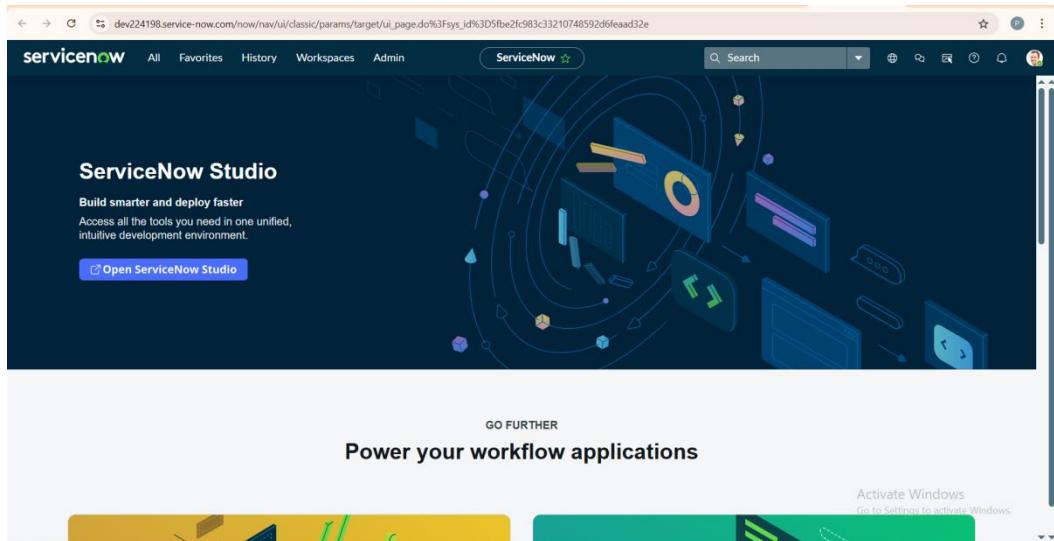
TASK INITIATION

Milestone 1 : 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.

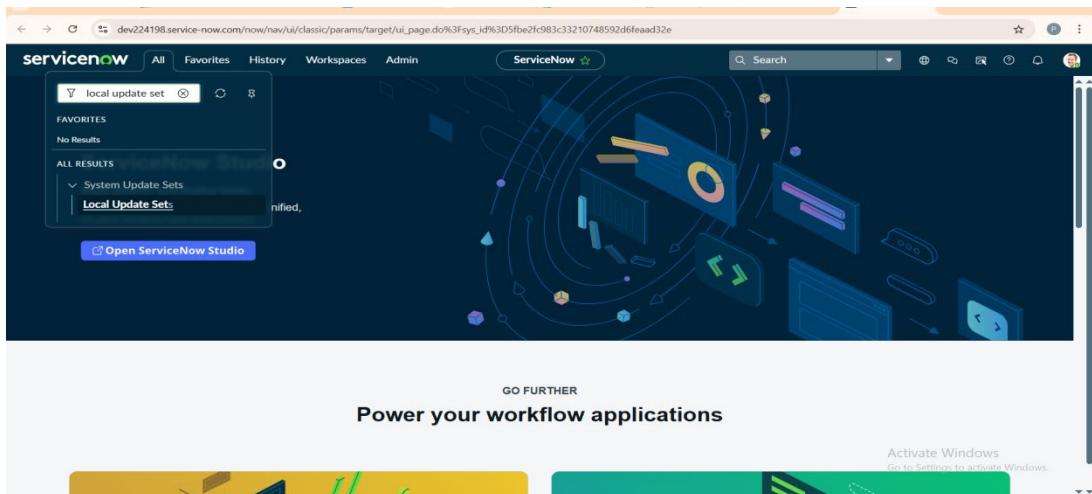
5. You'll receive an email with the instance details once it's ready.
6. Log in to your ServiceNow instance using the provided credentials.
7. Now you will navigate to the ServiceNow.



Milestone 2: 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.



The screenshot shows the ServiceNow Update Sets page. At the top, there's a navigation bar with links for All, Favorites, History, Workspaces, Admin, and a search bar. Below the navigation is a toolbar with icons for search, refresh, and other functions. The main area is titled "Update Sets" with a star icon. A sub-header says "Application Global". The form itself has fields for "Name" (set to "New Update Set"), "State" (set to "In progress"), "Parent" (empty), "Release date" (empty), and a large "Description" text area. At the bottom of the form are two buttons: "Submit" and "Submit and Make Current".

Milestone 3: Table(Family Expenses)

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

The screenshot shows the ServiceNow Table - New Record page. At the top, there's a navigation bar with links for All, Favorites, History, Admin, and a search bar. Below the navigation is a toolbar with icons for search, refresh, and other functions. The main area is titled "Table - New Record" with a star icon. A note at the top says "ServiceNow recommends creating custom tables in scoped applications. To learn more about creating scoped applications, click [here](#)". Below this is a message: "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 itself has fields for "Label" (set to "Family Expenses"), "Name" (set to "u_family_expenses"), "Extends table" (empty), "Application" (set to "Global"), "Create module" (checked), "Create mobile module" (checked), "Add module to menu" (set to "-- Create new --"), and "New menu name" (set to "Family Expenditure"). At the bottom are "Submit" and "Cancel" buttons.

Activity 2: Creation of Family Expenses Table

1. Near Columns Double click near insert a new row.
2. Give the details as:

The screenshot shows the ServiceNow Dictionary Entry Number View: Advanced screen. At the top, there are buttons for Delete Column and Update. Below that is a 'Display' checkbox. A note says 'Alters the behavior of a field or functionality that depends on the field.' with a 'More Info' link. The 'Default Value' tab is selected. It contains fields for 'Use dynamic default' (checked), 'Dynamic default value' (set to 'Get Next Padded Number'), and a 'Lookup using list' button. At the bottom are 'Delete Column' and 'Update' buttons.

Column label : Number

Type : String

3. Double click on insert a new row again.

4. Give the details as:

Column label : Date

Type : Date

5. Double click on insert a new row again

6. Give the details as:

Column label : Amount

Type : Integer

7. Double click on insert a new row again

8. Give the details as:

Column label : Expense Details

Type : String

Max length : 800

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

The screenshot shows the ServiceNow Table - Family Expenses screen. At the top, there are buttons for Delete, Update, and Delete All Records. Below that is a search bar and a 'New' button. The main area shows a table of 'Dictionary Entries'. The columns are: Column label, Type, Reference, Max length, Default value, and Display. The table contains the following data:

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

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
4. Click on Update.
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
8. Click on Submit

The screenshot shows the 'Number' maintenance screen in ServiceNow. At the top, there's a header with a back arrow, a 'Number' link, and a 'New record' button. Below the header are several input fields: 'Table' set to 'Family Expenses', 'Prefix' set to 'MFE', 'Number' (mandatory field) containing '1,000', 'Application' set to 'Global', and 'Number of digits' set to '7'. At the bottom left is a 'Submit' button, and below it are 'Related Links' for 'Show Counter'.

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.
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.

The screenshot shows the 'Form Design' interface for the 'Family Expenses [u_family]' table. On the left, there's a sidebar with tabs for 'Fields' (selected) and 'Field Types'. Under 'Fields', there are sections for 'Created', 'Updated', and 'Expense Details'. The main area shows a 'Form Design' with a '2 Column' layout. It contains three rows of fields: 'Number', 'Amount', and 'Date'. Each field has a gear icon to its right, which can be used to configure settings like 'Read Only' or 'Mandatory'. Below the main form area, there's a section labeled 'Expense Details'.

Milestone 4: Table(Daily Expenses)

Activity 1: Creation of Table(Daily Expenses)

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

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 click on insert a new row again
4. Give the details as:
Column label : Date
Type : Date
5. Double click on insert a new row again
6. Give the details as:
Column label : Expense
Type : Integer
7. Double click on insert a new row again
8. Give the details as:

The screenshot shows the ServiceNow 'Table - New Record' page. At the top, the URL is dev347226.service-now.com/nav/ui/classic/params/target/sys_db_object.do%3Fsys_id%3D-1%26sys_is_list%3Dtrue%26sys_target%3Dsys_db_object%26sy... . The page title is 'Table - New Record'. Below the title, it says 'Table New record'. A message from ServiceNow recommends creating custom tables in scoped applications. A blue info bar at the bottom states: '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.' Fields on the form include: Label (Daily Expenses), Application (Global), Name (u_daily_expenses), Create module (checked), Extends table (empty), Create mobile module (checked), Add module to menu (Family Expenditure), and Remote Table (unchecked). Buttons for 'Submit' and 'Cancel' are at the bottom right.

Column label : Family Member Name
Type : Reference
Max length : 800

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

9. Double click on insert a new row again

10. Give the details as:

Column label : Comments

Type : String

Max length : 800

11. 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

4. Click on Update.

Dictionary Entry
Number View: Advanced

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 [Lookup using list](#)

Dynamic default value

[Delete Column](#) [Update](#)

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

Number - New Record

All Favorites History : [Search](#)

Number New record

Table: Daily Expenses Prefix: DFE Number: 1,000 Application: Global Number of digits: 7

[Submit](#)

Related Links
[Show Counter](#)

Activity 4: Configure the Form

Form Design

Fields Field Types

Filter

Fields

- Created
- Created by
- Updated
- Updated by
- Updates

Formatters

- Activities (filtered)
- Contextual Search Results
- Ratings

Daily Expenses [u_daily_expenses]

Number	Date	Family Member Name	Expense
Comments			

2 Column

1 Column

Undo Save

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.
 5. Make Number Read-Only Field by clicking on the gear icon and checking Read-Only
 6. Make Date, Family Member Name Mandatory Field by clicking on the gear icon and checking Mandatory
 7. Click on Save.

Milestone 5: Creation of Relationship

Activity 1: Creation of Relationship between Family Expenses and Daily Expenses tables

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.

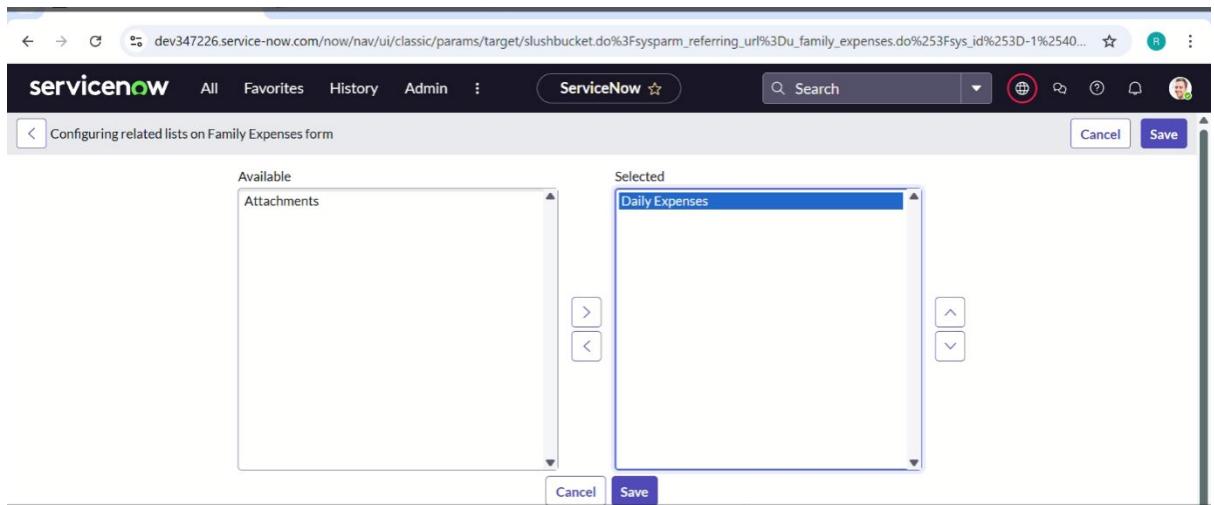
The screenshot shows the ServiceNow interface for creating a new relationship record. The top navigation bar includes links for All, Favorites, History, and a search bar labeled "Relationship - New Record". Below the header, there's a breadcrumb trail "Relationship" and a link "New record". The main form has fields for "Name" (set to "Daily Expenses"), "Application" (set to "Global"), and checkboxes for "Advanced" and "Query from table". The "Applies to table" dropdown is set to "Family Expenses [u_family_expe...]" and the "Queries from table" dropdown is set to "Daily Expenses [u_daily_expens...>". A note at the bottom left explains the purpose of the refined query script. The bottom section contains a code editor with a toolbar above it, showing a snippet of ECMAScript 2021 code:

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

Milestone 6: Configuring Related List on Family Expenses

Activity 1: Configuring Related List on Family Expenses

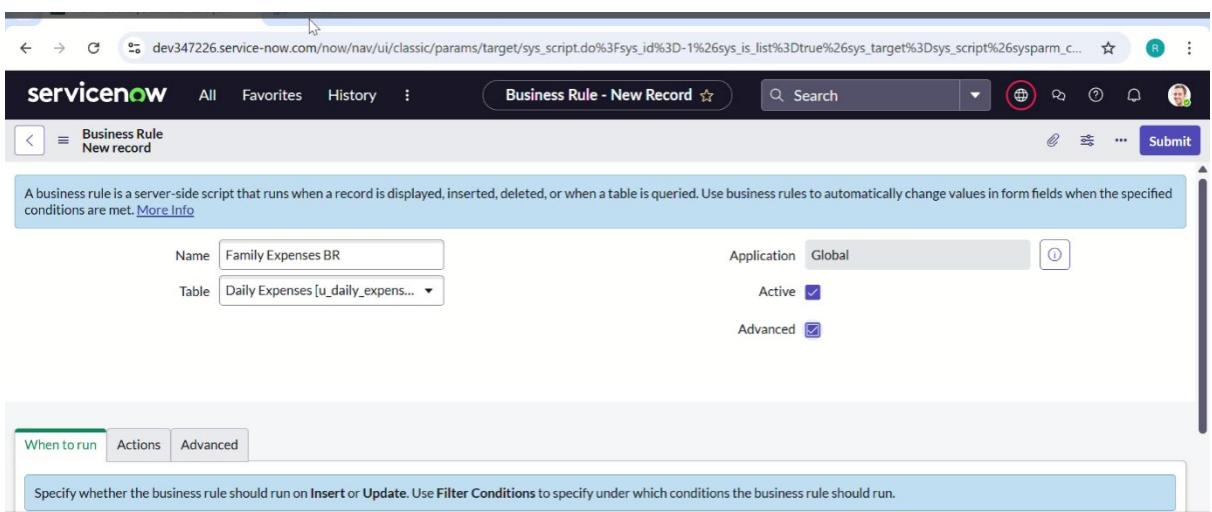
1. Go to All >> In the filter search for Family Expenses >> Open Family Expenses
2. Click on New
3. Go to the Header and right click there >> click on Configure >> Select Related Lists
4. Add Daily Expenses to the Selected Area.
5. Click on Save



Milestone 7: Business Rules

Activity 1: 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

Business Rule
New record

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: [Edit](#)

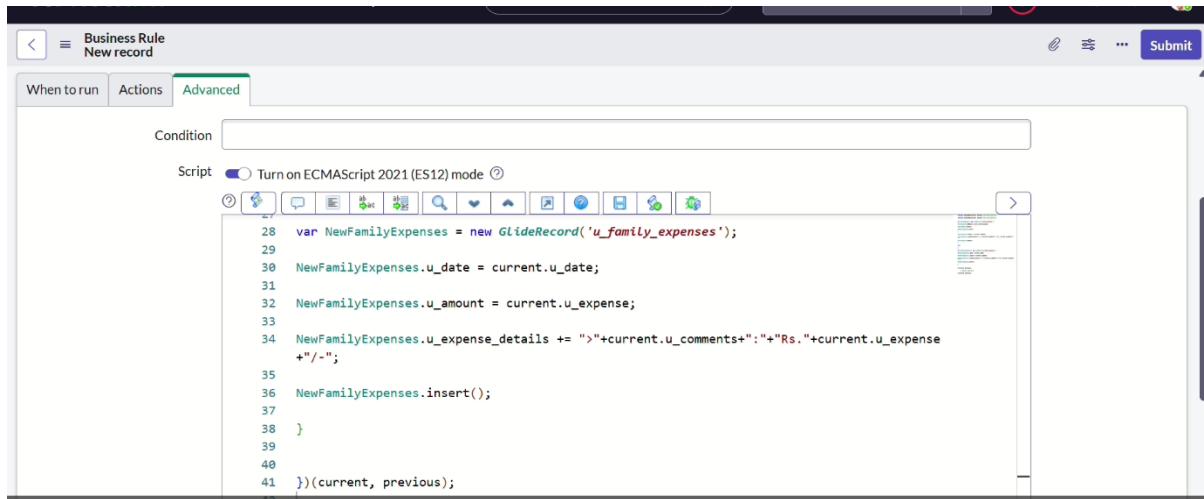
Submit

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

```
(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
        ">"+current.u_comments+":"+ "Rs." +current.u_expense+/-";
        FamilyExpenses.update();
    }
    else
    {
        varNewFamilyExpenses = 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);

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



```
Condition: 
Script: Turn on ECMAScript 2021 (ES12) mode
28 var NewFamilyExpenses = new GlideRecord('u_family_expenses');
29
30 NewFamilyExpenses.u_date = current.u_date;
31
32 NewFamilyExpenses.u_amount = current.u_expense;
33
34 NewFamilyExpenses.u_expense_details += ">" + current.u_comments + ":" + "Rs." + current.u_expense
+ "/-";
35
36 NewFamilyExpenses.insert();
37
38 }
39
40
41 })(current, previous);
```

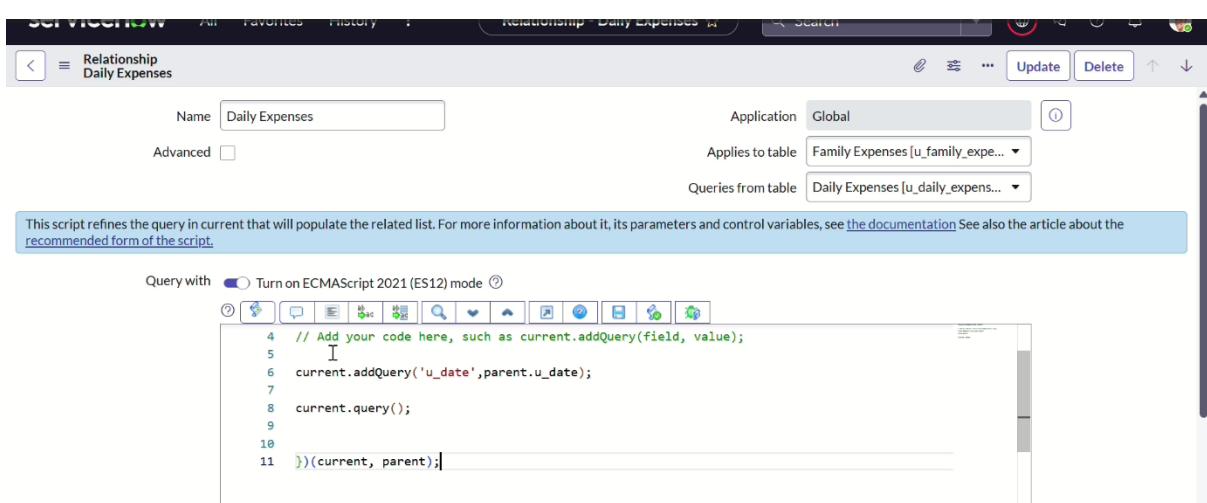
Milestone 8: Relationship

Activity 1: 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.

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

5. Click on Update.



Name: Daily Expenses
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);
```

