Calculating Family Expenses Using Servicenow

Team id: NM2025TMID13211

Team Members: 04

Team Leader: Prakash A

Team Member 1: Shanmugapriyan S

Team Member 2: Dinesh Kumar S

Team Member 3: Suresh K

Objective: To build a ServiceNow-based expense management app that records, categorizes, and analyzes family expenses with dashboards and alerts.

Skills: ServiceNow app development, Flow Designer, reporting & dashboards, JavaScript, database design, and basic UI/UX.

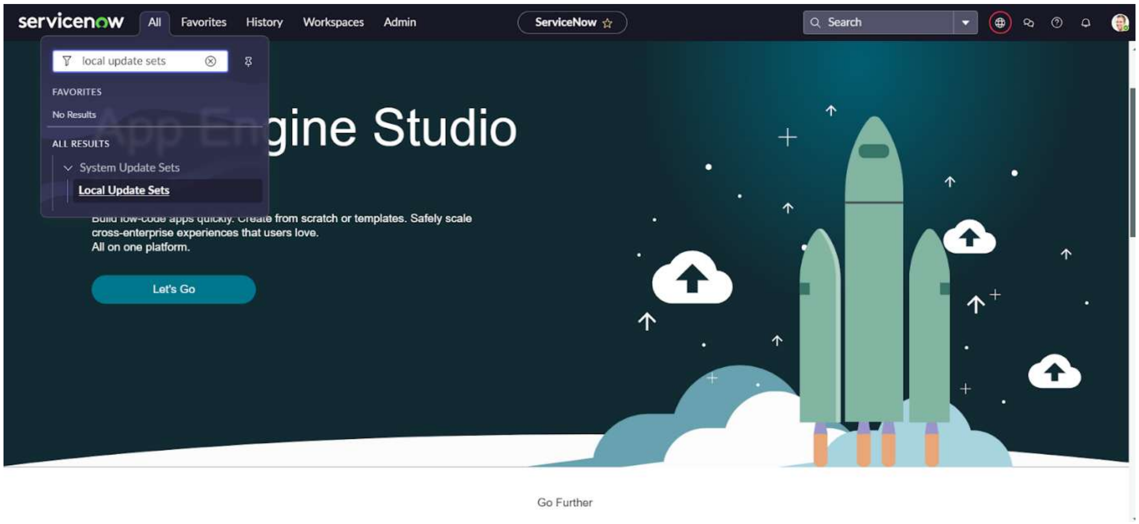
TASK INITIATION

Milestone 1 :Users

Activity 1: Create Users

Creation of New Update Set

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

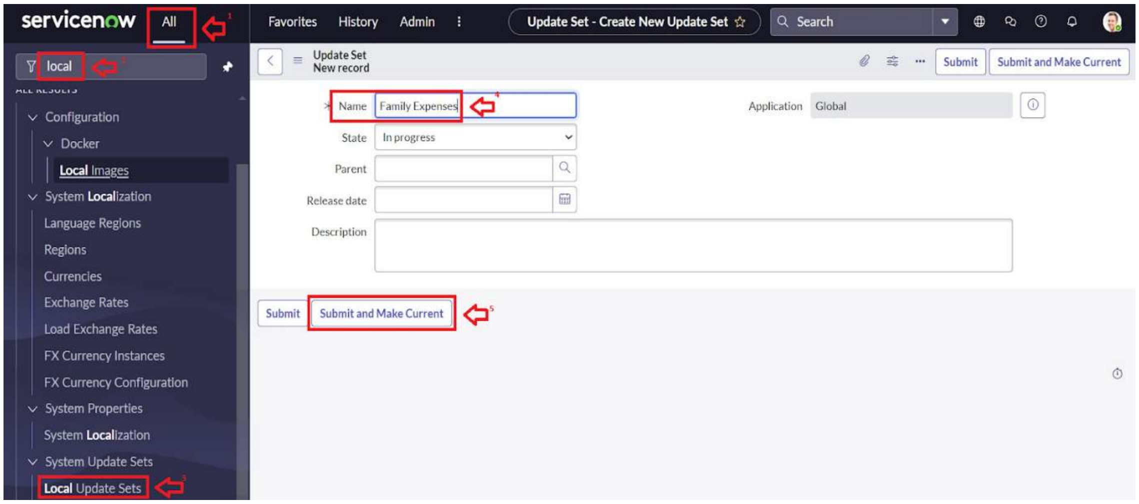


Build low-code apps quickly. Create from scratch or templates. Safely scale cross-enterprise experiences that users love. All on one platform.

2. Enter the Details as:

Name : Family Expenses

3. Then click on Submit and Make current.



Name: Family Expenses

State: In progress

Parent:

Release date:

Description:

Application: Global

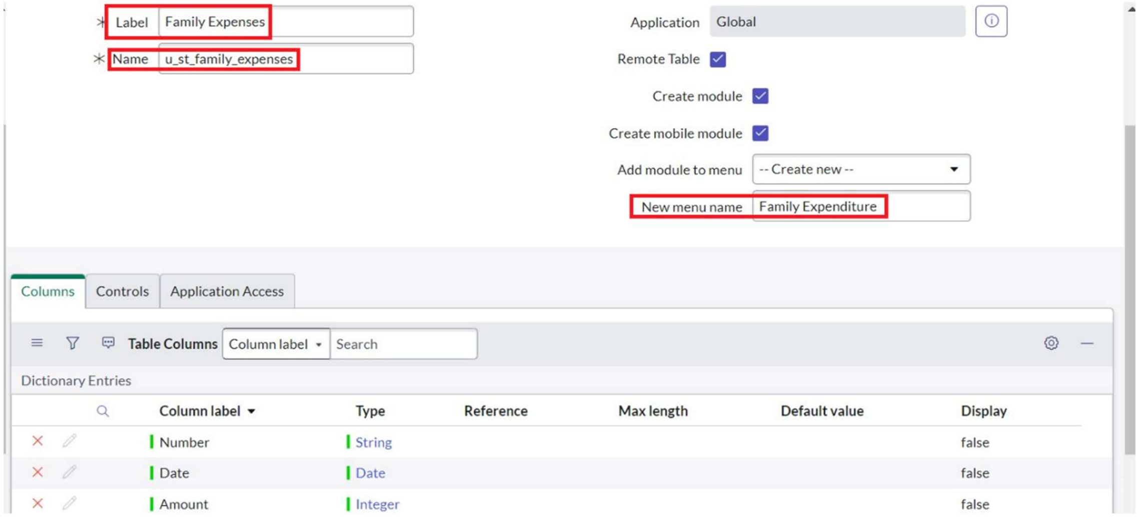
Creation of Family Expenses Table

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



* Label	Family Expenses
* Name	u_st_family_expenses
Application	Global
Remote Table	☑
Create module	☑
Create mobile module	☑
Add module to menu	-- Create new --
New menu name	Family Expenditure

Column label	Type	Reference	Max length	Default value	Display
Number	String				false
Date	Date				false
Amount	Integer				false

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

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 : Amount

Type : Integer

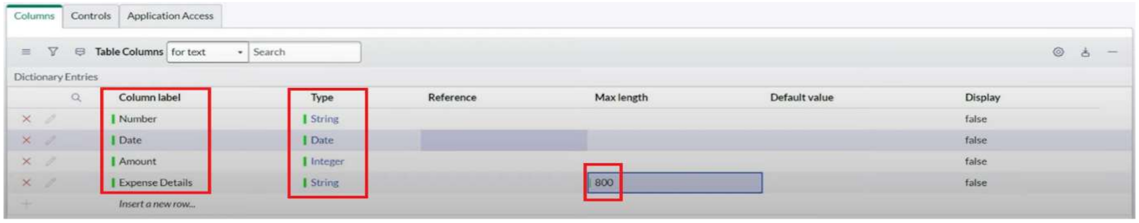
7. Double click on insert a new row again

8. Give the details as:

Column label : Expense Details

Type : String

Max length : 800



Column label	Type	Reference	Max length	Default value	Display
Number	String				false
Date	Date				false
Amount	Integer				false
Expense Details	String		800		false
Insert a new row...					

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

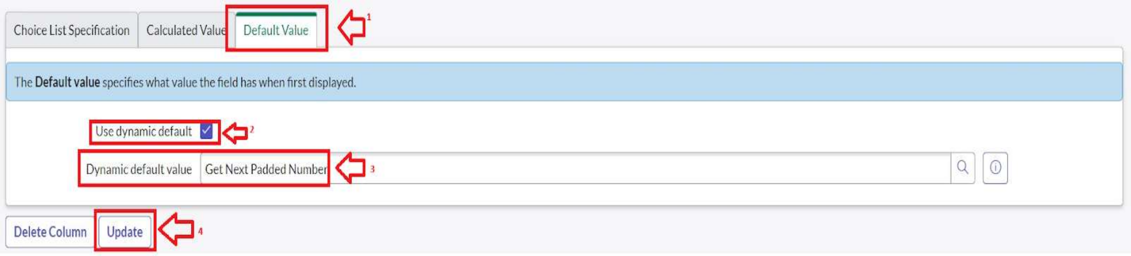
Making Number Field an Auto-Number

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.



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

Use dynamic default ☑

Dynamic default value Get Next Padded Number

5.

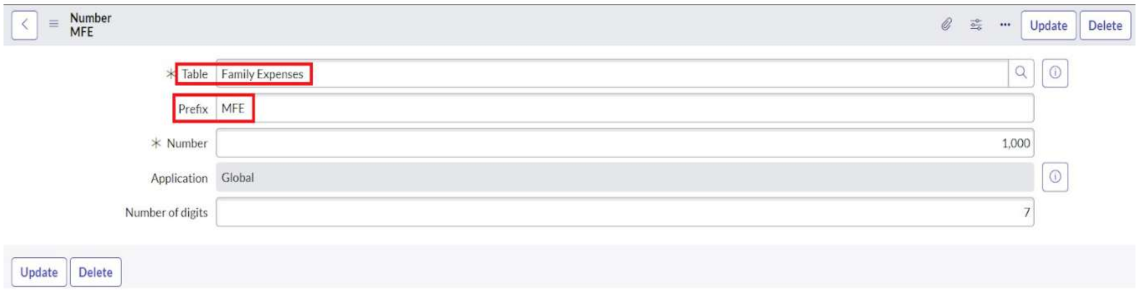
6. Go to All >> In the filter search for Number Maintenance >> select Number Maintenance

7. Click on New.

8. Enter the below Details:

Table : Family Expenses

Prefix : MFE

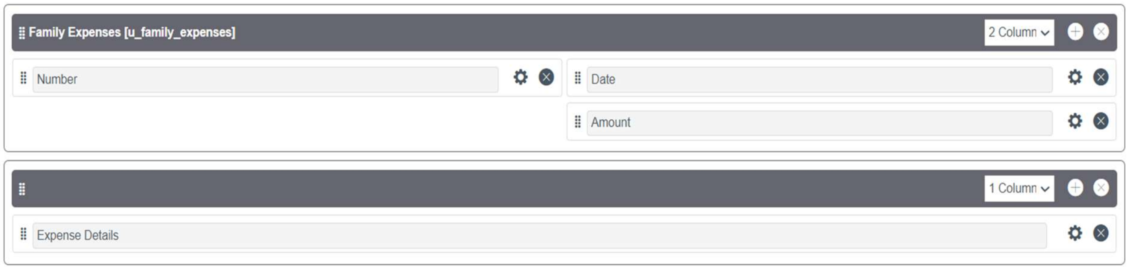


* Table	Family Expenses
Prefix	MFE
* Number	1,000
Application	Global
Number of digits	7

9. Click on Submit.

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.

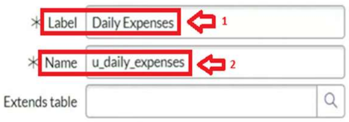
Creation of Daily Expenses Table

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

Label : Daily Expenses

Name : Auto-Populated

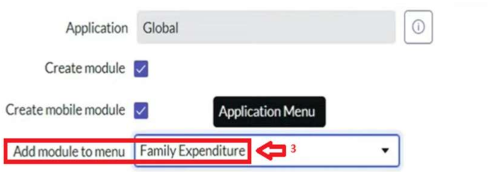
Add Module to menu : Family Expenditure



*Label Daily Expenses

*Name u_daily_expenses

Extends table



Application Global

Create module ☑

Create mobile module ☑

Add module to menu Family Expenditure

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

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:

Column label : Family Member Name

Type : Reference

Max length : 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.

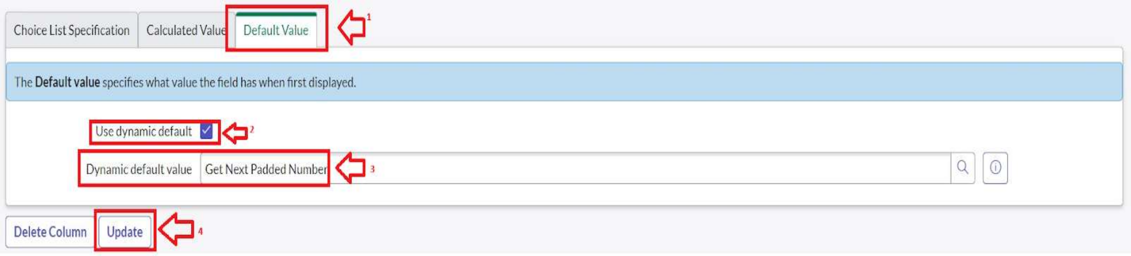
Making Number Field an Auto-Number

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.



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

Use dynamic default ☑

Dynamic default value Get Next Padded Number

5.

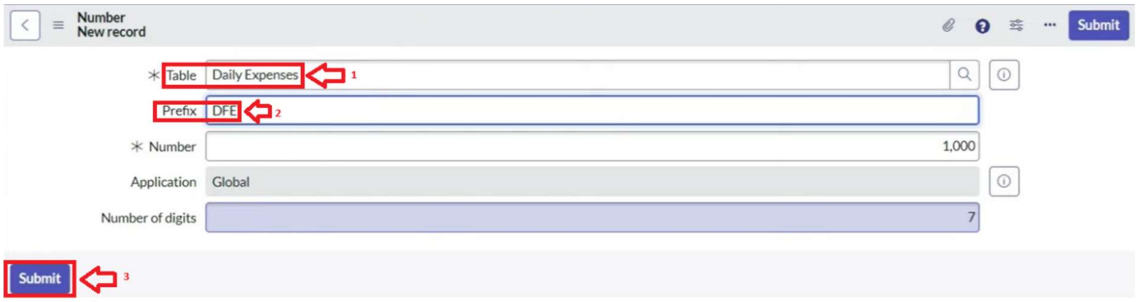
6. Go to All >> In the filter search for Number Maintenance >> select Number Maintenance

7. Click on New.

8. Enter the below Details:

Table : Family Expenses

Prefix : MFE

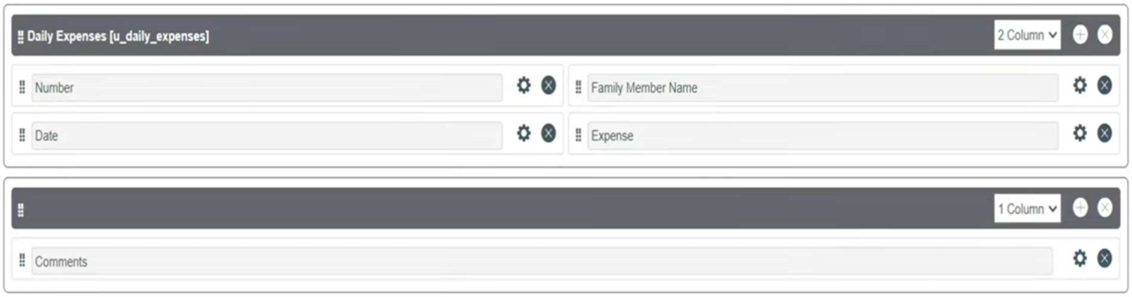


* Table	Daily Expenses
Prefix	DFE
* Number	1,000
Application	Global
Number of digits	7

9. Click on Submit.

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.



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.

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

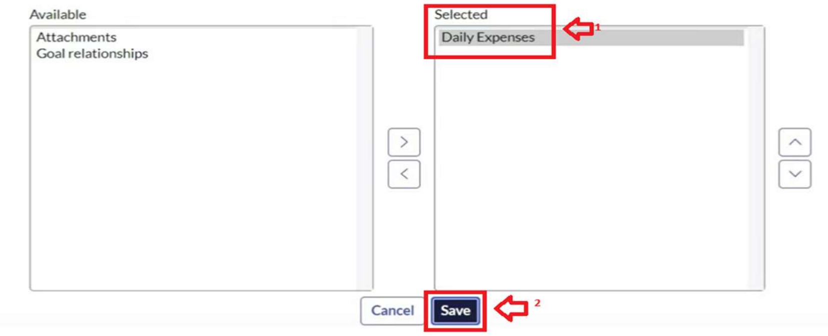
Applies to table : Select Family Expenses

Daily Expenses : Select Daily Expenses

4. Click Save.

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



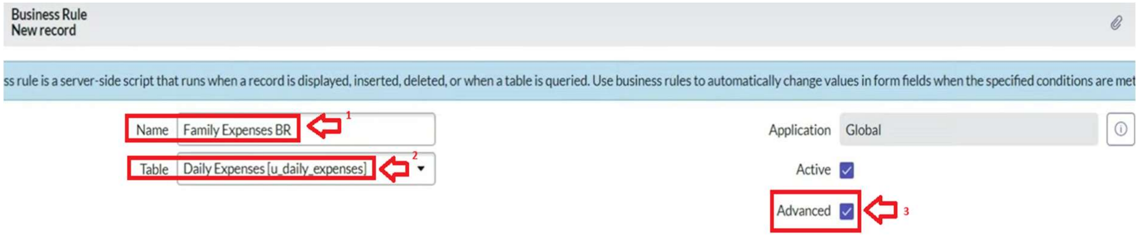
Creation of Business Rules

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



Business Rule
New record

ss rule is a server-side script that runs when a record is displayed, inserted, deleted, or when a table is queried. Use business rules to automatically change values in form fields when the specified conditions are met

Name: Family Expenses BR

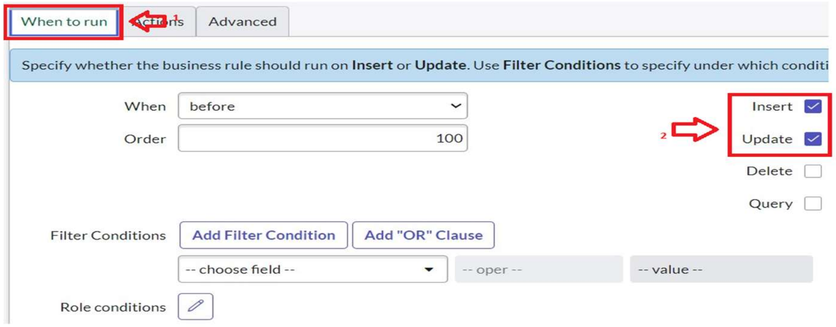
Table: Daily Expenses [u_daily_expenses]

Application: Global

Active ☑

Advanced ☑

4. In when to run Check Insert and Update



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

When: before

Order: 100

Insert ☑

Update ☑

Delete ☐

Query ☐

Filter Conditions: Add Filter Condition | Add "OR" Clause

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

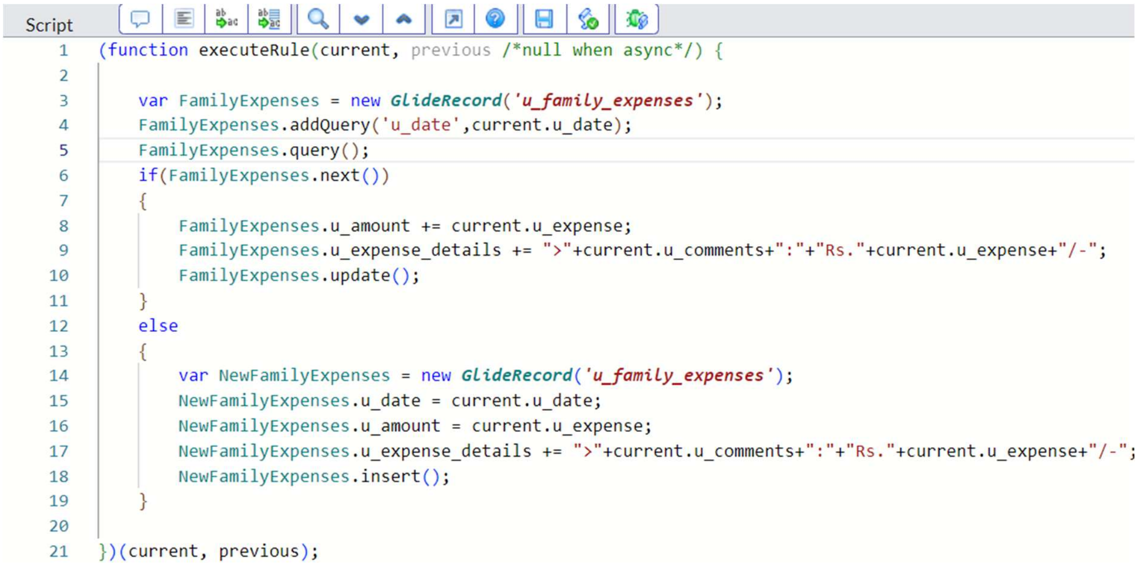
Role conditions

5. 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);
```



```
(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);
```

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

Configure the Relationship

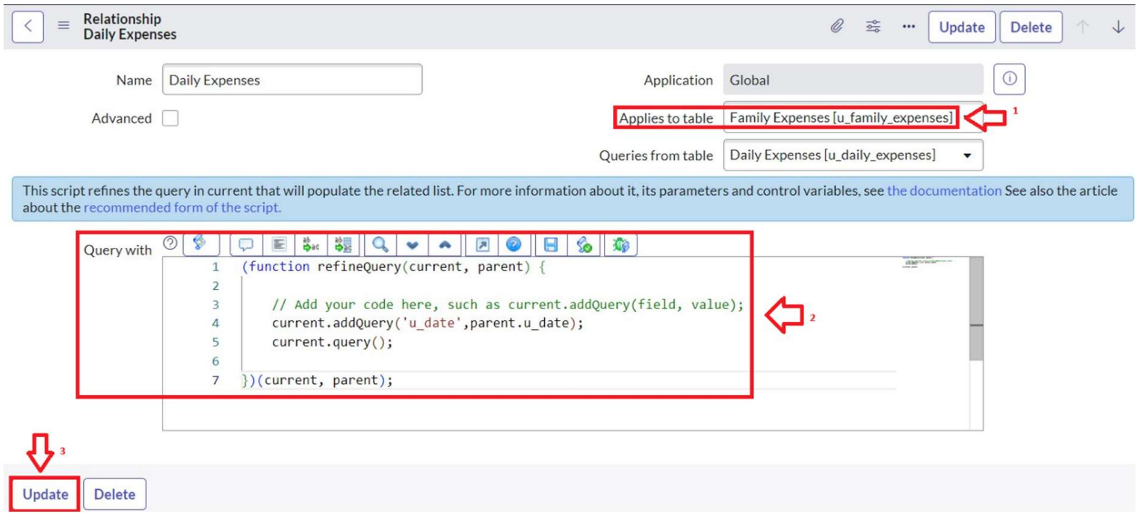
1. Go to All >> In the filter search for Relationships >> Open Relationships.
2. In that, open Daily Expenses Relationship.
3. For Applies to table : Select Family Expenses.
4. In Query with : write the below Query.

```
(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);
```

5. Click on Update.



Relationship
Daily Expenses

Name: Daily Expenses

Advanced □

Application: Global

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

```
(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);
```

Conclusion : Whether for business or personal use, servicenow proves to be powerful tool for structured expense management.