

# **Calculating Family Expenses Using Servicenow**

**Team id: NM2025TMID13210**

**Team Members: 04**

**Team Leader: NITHISHKUMAR P**

**Team Member 1: RAJESH G**

**Team Member 2: RAJESH M**

**Team Member 3: KARTHIK S**

**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.

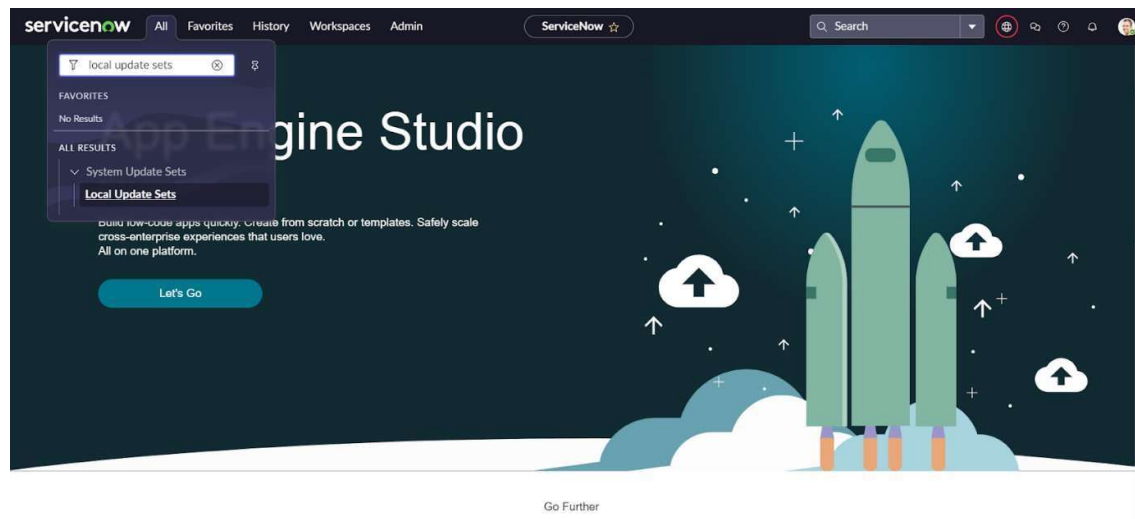
**TASK INITIATION**

## Milestone 1 :Users

### Activity 1: Create Users

#### Creation of New Update Set

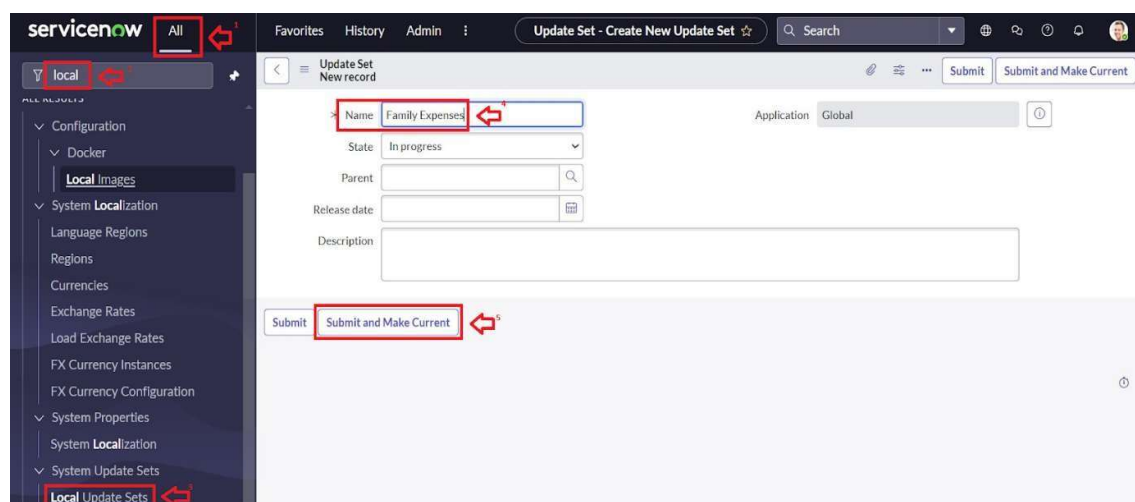
1. Go to All >> In the filter search for Local Update set > click on New.



2. Enter the Details as:

Name : Family Expenses

3. Then click on Submit and Make current.



#### 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

Columns Controls Application Access

Table Columns Column label Search

Dictionary Entries

	Column label	Type	Reference	Max length	Default value	Display
<input checked="" type="checkbox"/>	Number	String				false
<input checked="" type="checkbox"/>	Date	Date				false
<input checked="" type="checkbox"/>	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

Table Columns					
Dictionary Entries	Column label	Type	Reference	Max length	Default value
	Number	String			false
	Date	Date			false
	Amount	Integer			false
	Expense Details	String		800	false

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.

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**

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

The screenshot shows a form for creating a new Number Maintenance record. The form is titled "Number MFE" and has a header bar with "Update" and "Delete" buttons. The form fields are as follows:

- Table:** Family Expenses (highlighted with a red box)
- Prefix:** MFE (highlighted with a red box)
- \* Number:** 1,000
- Application:** Global
- Number of digits:** 7

At the bottom of the form, there are "Update" and "Delete" buttons.

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.

The screenshot shows a form editor for 'Family Expenses [u\_family\_expenses]'. The form is divided into two columns. The first column contains a 'Number' field. The second column contains 'Date' and 'Amount' fields. Each field has a gear icon for configuration. Below this, there is a section for 'Expense Details' with a single 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.

## 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   1  
 \* Name   2  
 Extends table  

Application    
 Create module ☒  
 Create mobile module ☒   
 Add module to menu   3

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.

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**

### 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

< ≡ Number New record

\* Table Daily Expenses

Prefix DFE

\* Number 1,000

Application Global

Number of digits 7

**Submit**

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.

The screenshot shows a form configuration window for 'Daily Expenses [u\_daily\_expenses]'. The form is organized into two columns. The first column contains 'Number' and 'Date' fields. The second column contains 'Family Member Name' and 'Expense' fields. Each field has a gear icon to its right, indicating it can be configured. Below the two-column section, there is a single-column section titled 'Comments' with a single text area field. The top right of the window shows '2 Column' and '1 Column' options, suggesting a toggle between different layouts.

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

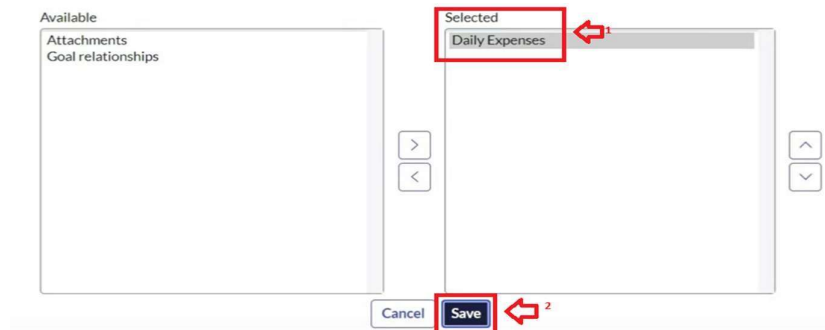
Applies to table : Select Family Expenses

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

Business 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	← <sup>1</sup>	Application	Global	ⓘ
Table	Daily Expenses (u daily expenses)	← <sup>2</sup>	Active	<input checked="" type="checkbox"/>	
			Advanced	<input checked="" type="checkbox"/>	← <sup>3</sup>

4. In when to run Check Insert and Update

When to run **When to run** Advanced

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

When

Order

Insert ☒

Update ☒

Delete ☐

Query ☐

Filter Conditions

-- choose field --

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;

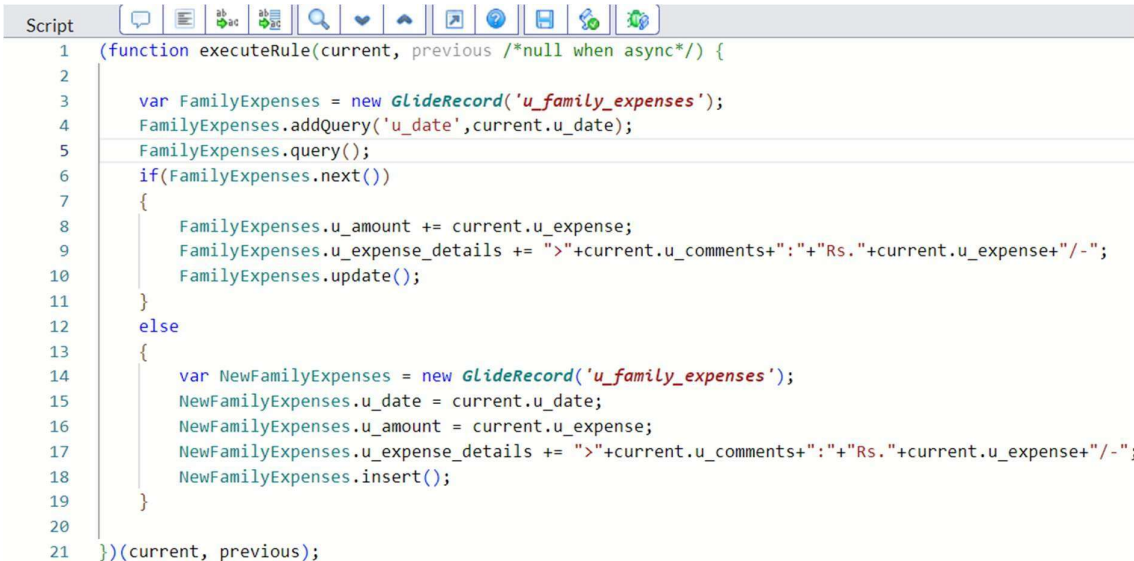
NewFamilyExpenses.u_expense_details +=
">" + current.u_comments + ":" + "Rs." + current.u_expense + "/-";

NewFamilyExpenses.insert();

}

})(current, previous);

```



The screenshot shows a script editor with a toolbar at the top containing icons for comments, undo, redo, search, and other standard development tools. The script is written in JavaScript and implements a logic to either update an existing record or insert a new one based on the presence of a 'u\_date' field.

```

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);

```

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

UpdateDelete

NameDaily Expenses

ApplicationGlobal

Advanced☐

Applies to tableFamily Expenses [u\_family\_expenses]

Queries from tableDaily 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

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

UpdateDelete

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