

# **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.

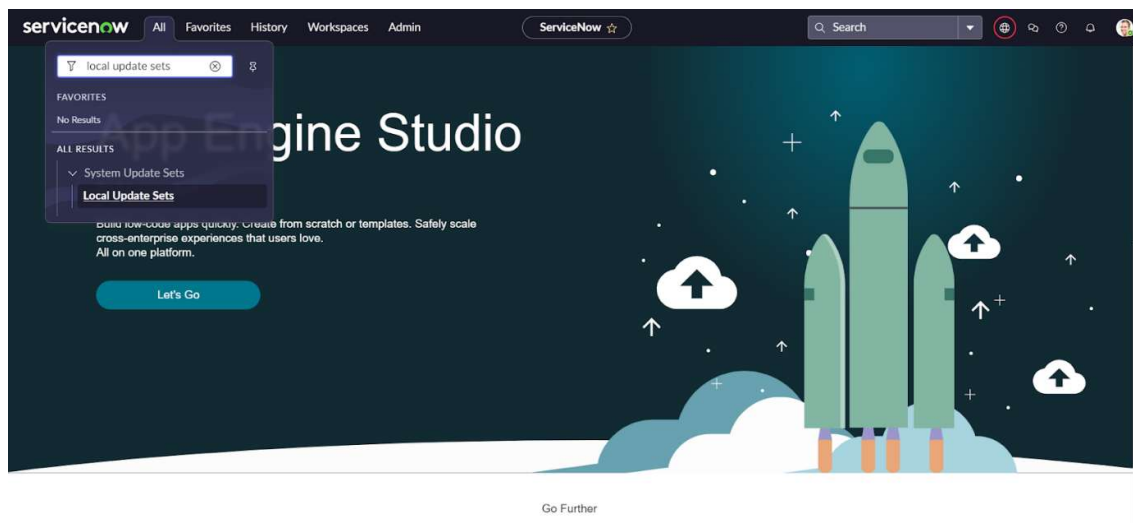
# TASK INITIATION

## Milestone 1 :Users

### Activity 1: Create Users

#### Creation of New Update Set

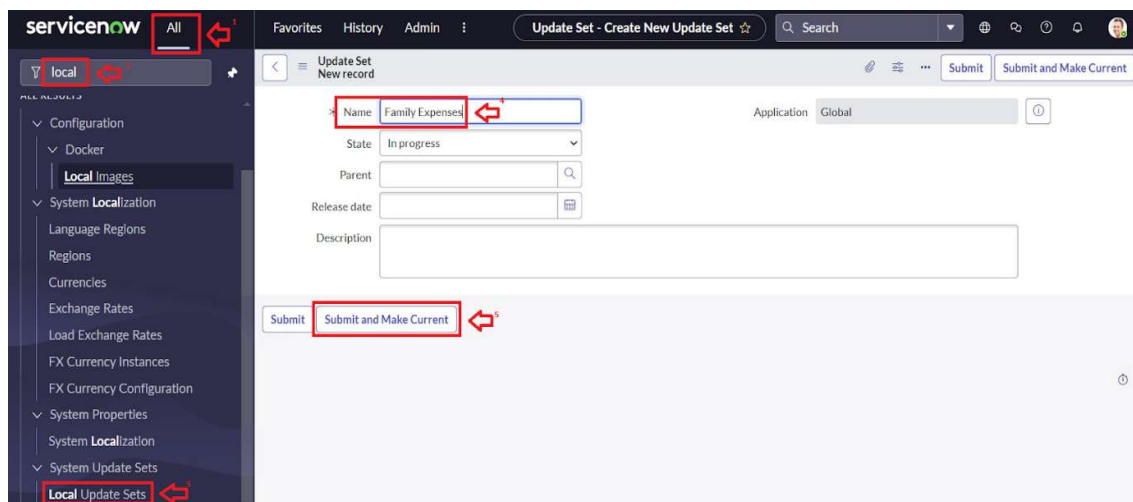
1. Go to All >> In the filter search for Local Update set > click on New.



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

The screenshot shows a web application interface for creating a new table. At the top, there are two input fields: 'Label' with the value 'Family Expenses' and 'Name' with the value 'u\_st\_family\_expenses'. To the right, there are several settings: 'Application' set to 'Global', 'Remote Table' checked, 'Create module' checked, 'Create mobile module' checked, 'Add module to menu' set to '-- Create new --', and 'New menu name' set to 'Family Expenditure'. Below these fields, there are three tabs: 'Columns', 'Controls', and 'Application Access'. The 'Columns' tab is active, showing a 'Table Columns' section with a search bar and a 'Dictionary Entries' table. The table has columns for 'Column label', 'Type', 'Reference', 'Max length', 'Default value', and 'Display'. It contains three entries: 'Number' (String), 'Date' (Date), and 'Amount' (Integer).

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

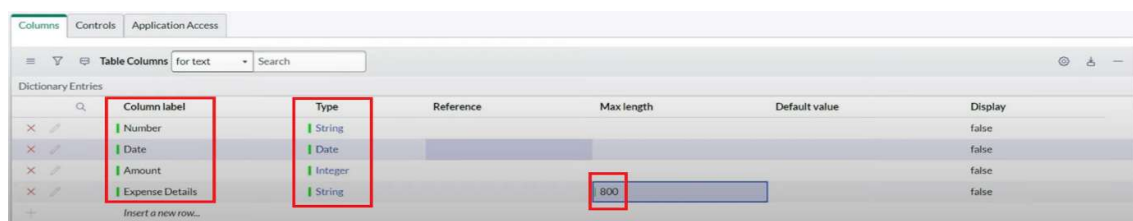
7. Double click on insert a new row again

8. Give the details as:

Column label : Expense Details

Type : String

Max length : 800



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** | [Red Arrow]

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 MFE

Table: Family Expenses

Prefix: MFE

\* Number: 1,000

Application: Global

Number of digits: 7

Update Delete

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.

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.

## 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 ☒

Application Menu

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 |

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 SAP 'New record' form for a table named 'Daily Expenses'. The form has several fields: 'Table' (set to 'Daily Expenses'), 'Prefix' (set to 'DFE'), 'Number' (set to '1,000'), 'Application' (set to 'Global'), and 'Number of digits' (set to '7'). A red box highlights the 'Table' field with an arrow pointing to it labeled '1'. Another red box highlights the 'Prefix' field with an arrow pointing to it labeled '2'. A third red box highlights the 'Submit' button at the bottom left with an arrow pointing to it labeled '3'.

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 image shows a screenshot of a web form titled "Daily Expenses [u\_daily\_expenses]". The form has a header bar with the title and a "2 Column" dropdown. Below the header, there are four input fields arranged in a 2x2 grid: "Number", "Date", "Family Member Name", and "Expense". Each field has a gear icon to its right, indicating configuration options. Below this grid is a "Comments" section with a single text input field and a gear icon. The form is styled with a light gray background and dark text.

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

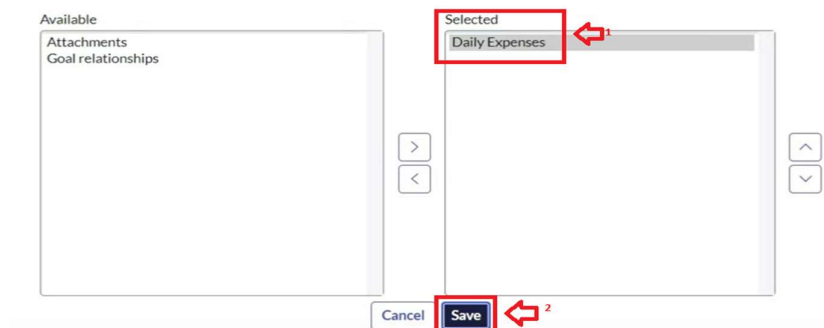
Applies to table : Select Family Expenses

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

When to run   Advanced

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

When before 

Order 100


Insert ☒ 


Update ☒

Delete ☐

Query ☐

Filter Conditions

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

Role conditions 

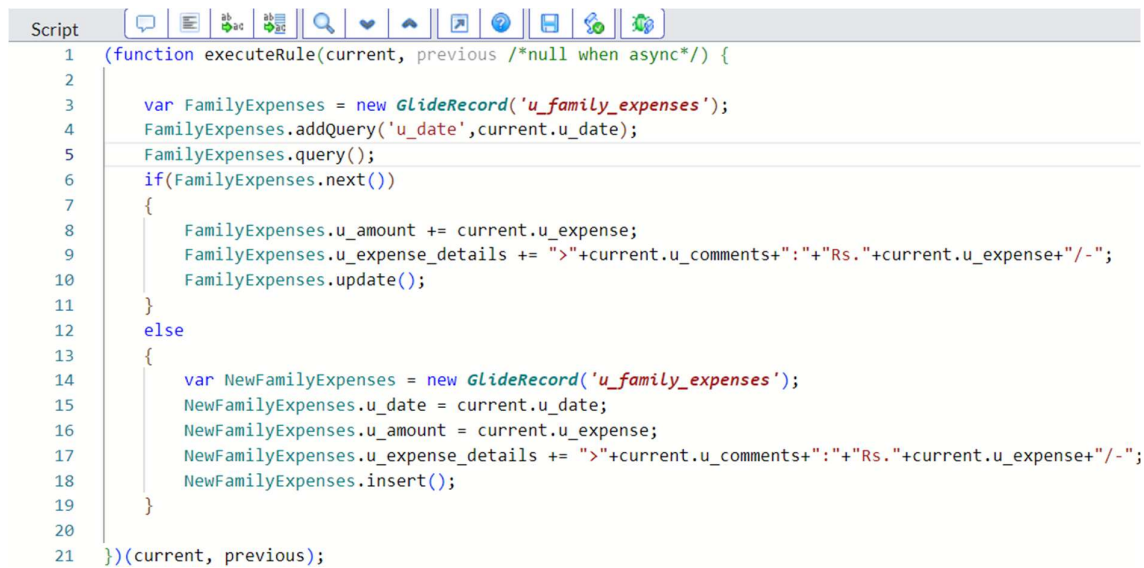
#### 5. In Advance(we write the code): Write the below code >>

```
(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);
```



```

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.

The screenshot shows the 'Refine Query' interface in ServiceNow. At the top, the 'Name' field is set to 'Daily Expenses' and the 'Application' is 'Global'. The 'Applies to table' dropdown is set to 'Family Expenses [u\_family\_expenses]', highlighted with a red box and a red arrow labeled '1'. Below this, the 'Queries from table' dropdown is set to 'Daily Expenses [u\_daily\_expenses]'. A blue informational banner states: '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](#).' The 'Query with' section contains a code editor with the following script: 

```
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);
```

 The code editor is highlighted with a red box and a red arrow labeled '2'. At the bottom, the 'Update' button is highlighted with a red box and a red arrow labeled '3'.

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