

## Performance and Testing

<b>Date</b>	30/10/2025
<b>Time id</b>	NM2025TMID05890
<b>Project name</b>	Calculating Family Expenses using Service Now
<b>Maximum mark</b>	4 marks

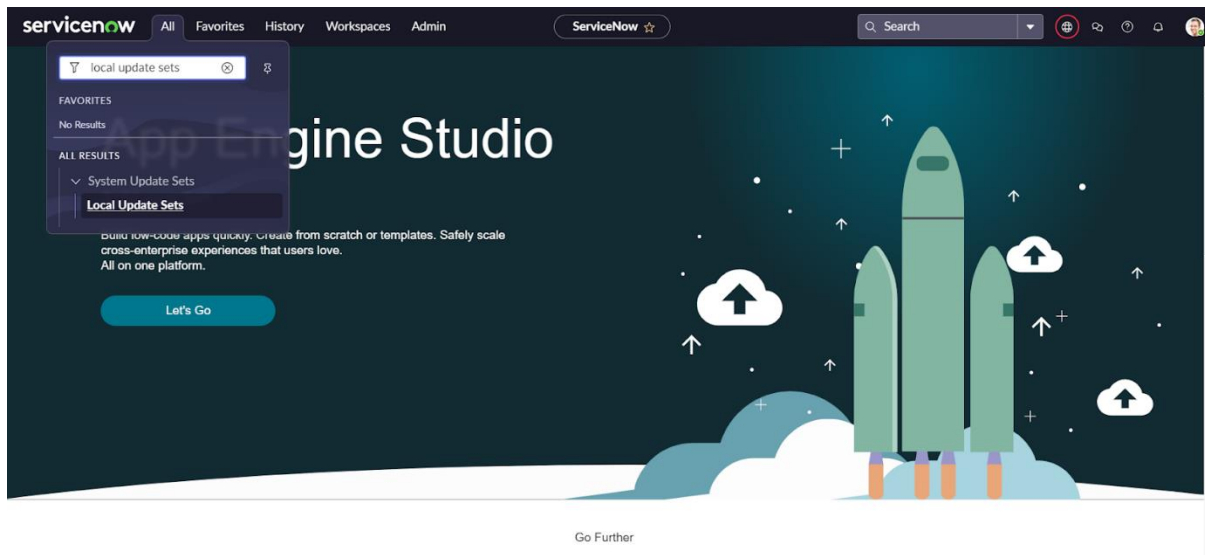
### Model performance testing

#### 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.

### Creation of New Update Set

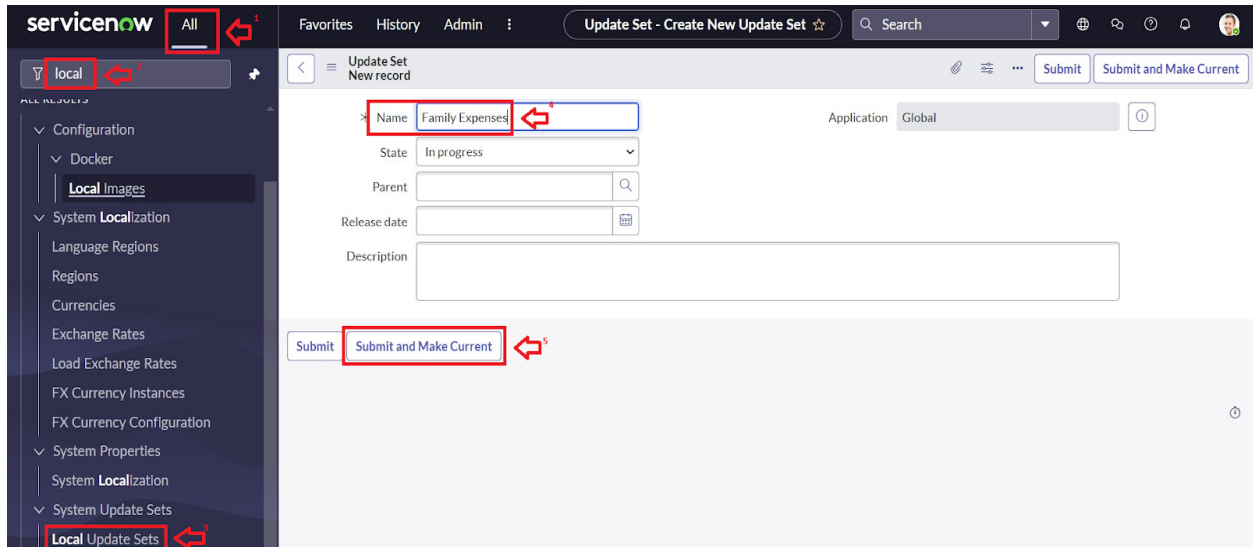
1. Go to All >> In the filter search for Local Update set > click on New.



2. Enter the Details as:

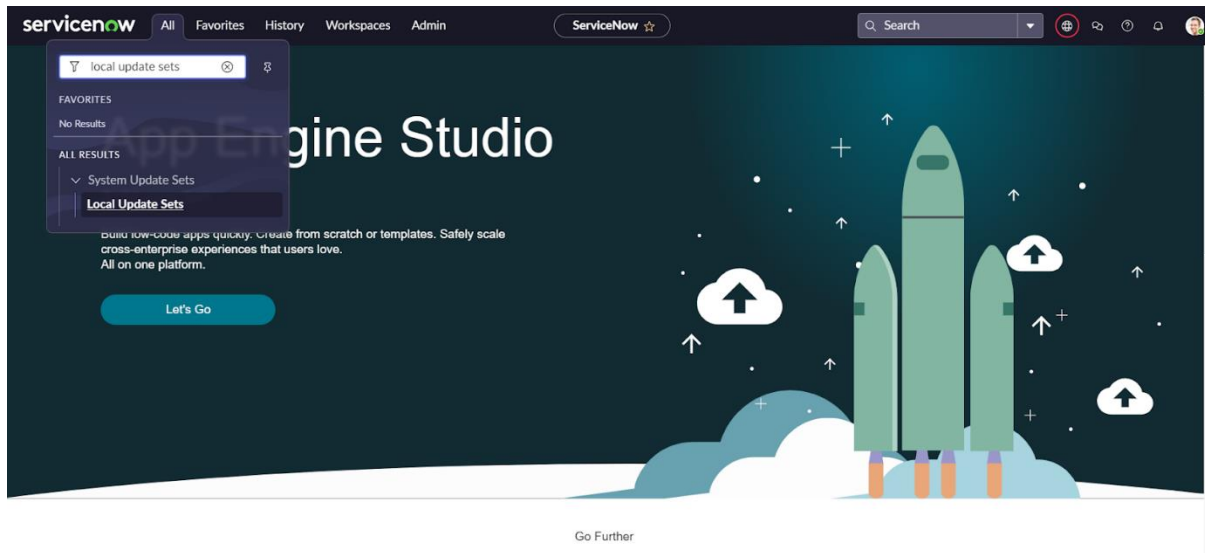
Name : Family Expenses

3. Then click on Submit and Make current.



## Creation of Table

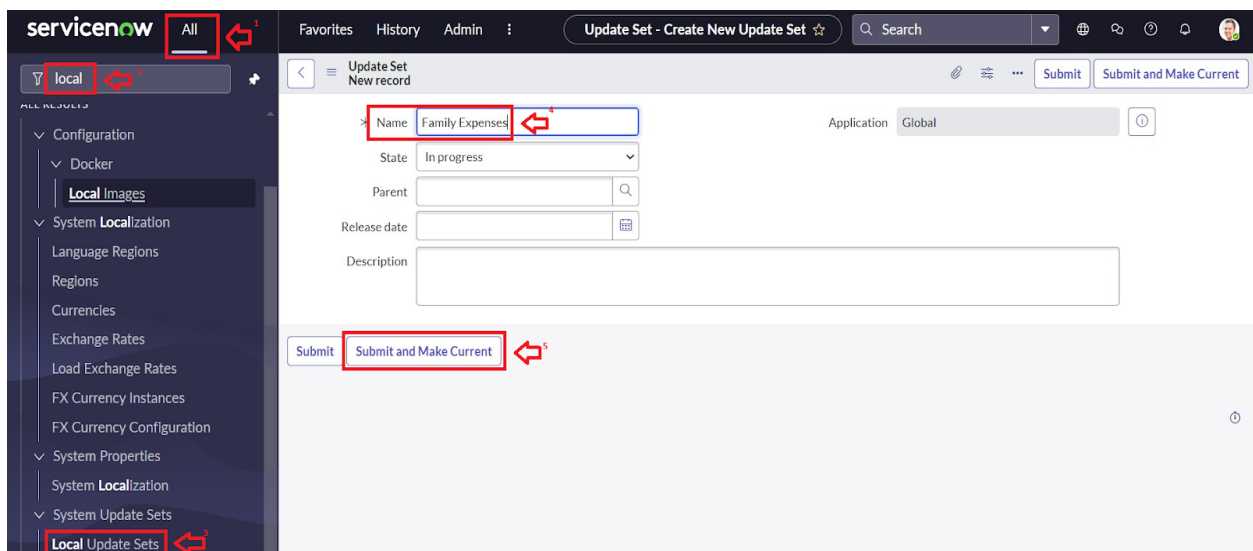
1. Go to All >> In the filter search for Local Update set > click on New.



2. Enter the Details as:

Name : Family Expenses

3. Then click on Submit and Make current.



## Creation of Table

### 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 form for creating a new table. The form includes the following fields and options:

- 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

Below the form, there is a section titled 'Columns' with a tabbed interface. The 'Columns' tab is active, showing a table of dictionary entries:

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

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** 

 **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

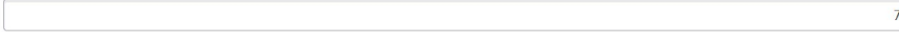
< ≡ Number MFE   ...  **Update** **Delete**



\* Table **Family Expenses**  

Prefix **MFE**

\* Number  1,000

Application Global 

Number of digits  7

8.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 displays a form configuration window for 'Family Expenses [u\_family\_expenses]'. The top bar indicates a '2 Column' layout. The main area contains three input fields: 'Number', 'Date', and 'Amount'. Each field has a gear icon to its right, indicating configuration options. Below this, there is a section for 'Expense Details' in a '1 Column' layout, also with a gear icon. The interface is clean and modern, with a dark header bar and light gray input fields.

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.

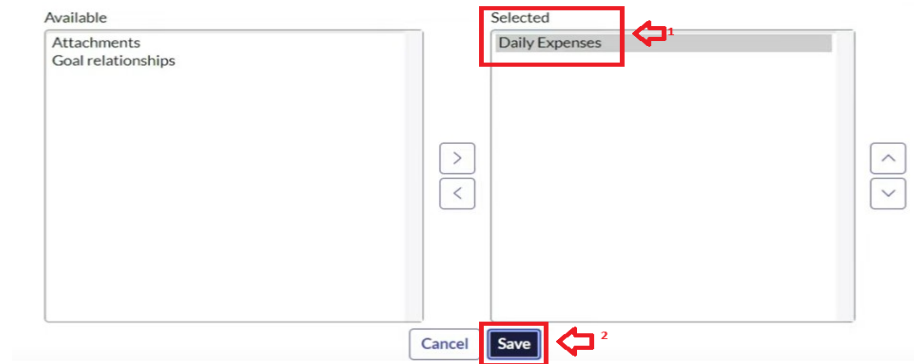


## **Creation of Relationship between Family Expenses and Daily Expenses tables**

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  


Table  

Application  

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

Order

 **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;
```

```

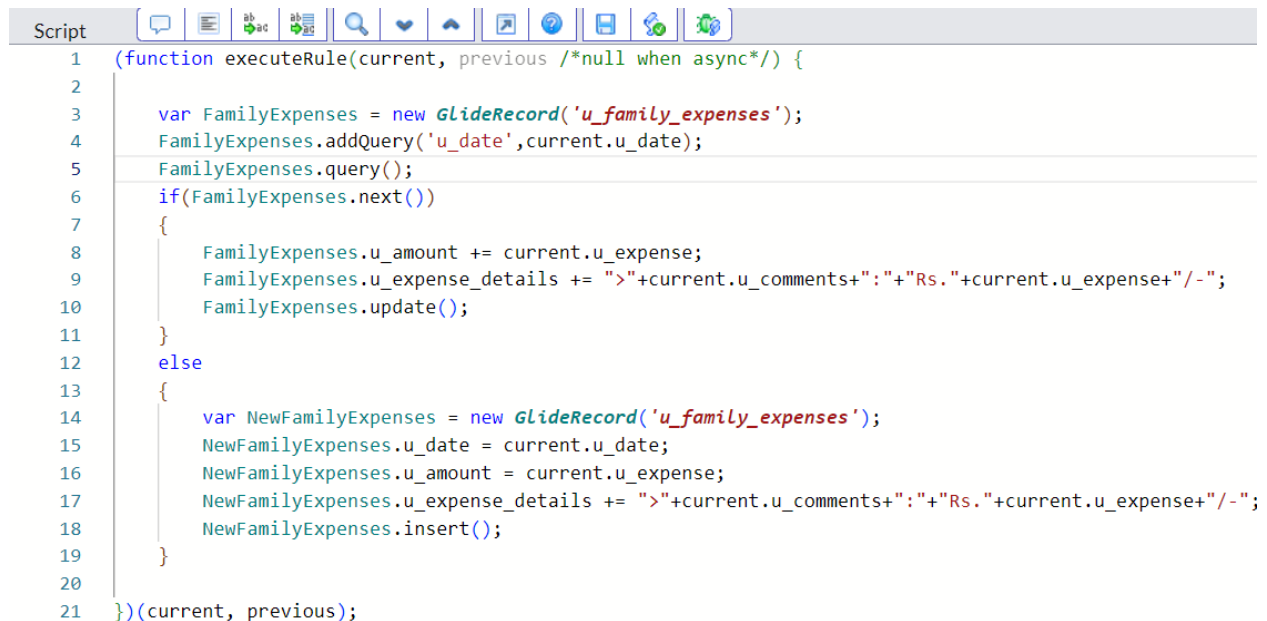
NewFamilyExpenses.u_expense_details +=
">" + current.u_comments + ":" + "Rs." + current.u_expense + "/-";

NewFamilyExpenses.insert();

}

})(current, previous);

```



The screenshot shows a script editor with a toolbar at the top. The script is as follows:

```

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 a web form titled "Relationship Daily Expenses". At the top, there are buttons for "Update" and "Delete". Below the title, there are input fields for "Name" (containing "Daily Expenses") and "Application" (containing "Global"). An "Advanced" checkbox is present. A red box labeled "1" highlights the "Applies to table" dropdown, which is set to "Family Expenses [u\_family\_expenses]". 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." Below the banner is a "Query with" section containing a code editor with a red box labeled "2" around it. The code is a JavaScript function: 

```
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);
```

 At the bottom, there are "Update" and "Delete" buttons, with a red box labeled "3" around the "Update" button.

## Conclusion

Calculating family expenses is fundamental for maintaining financial stability and achieving long-term goals. Accurate tracking of expenses allows families to understand their spending habits, identify unnecessary costs, and allocate funds more effectively across needs, wants, and savings. Utilizing tools or digital systems to automate this process reduces errors and saves time, enhancing transparency and collaboration among family members. Implementing a well-structured budgeting method, such as the 50/30/20 rule or envelope budgeting, empowers families to balance essential expenses

with discretionary spending and savings. Ultimately, regular and thoughtful expense calculation fosters informed decision-making, reduces financial stress, and supports a secure and prosperous household financial future.