

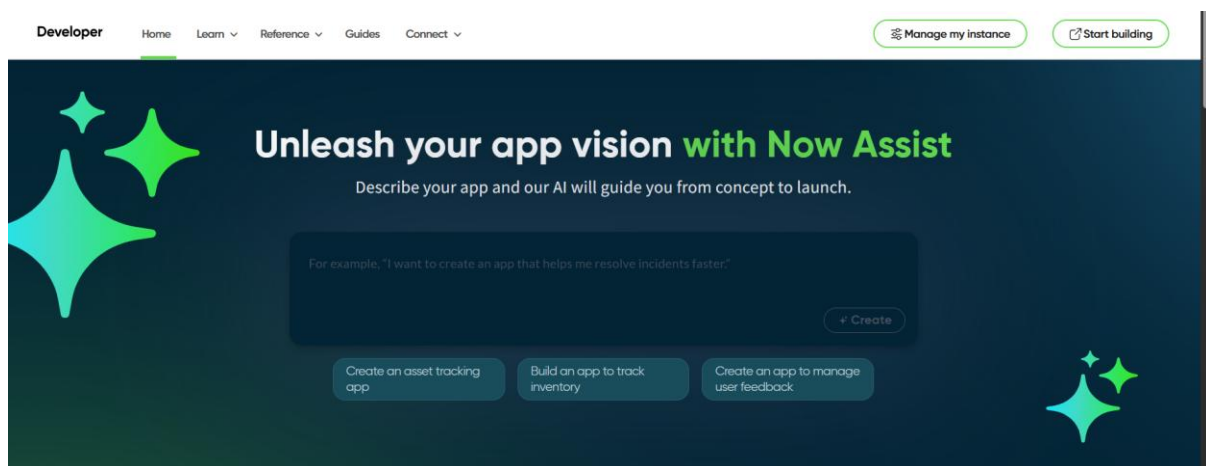
# Calculating Family Expenses using Service Now

## 1. Setting up ServiceNow Instance

Sign up for a developer account at [developer.servicenow.com](https://developer.servicenow.com).

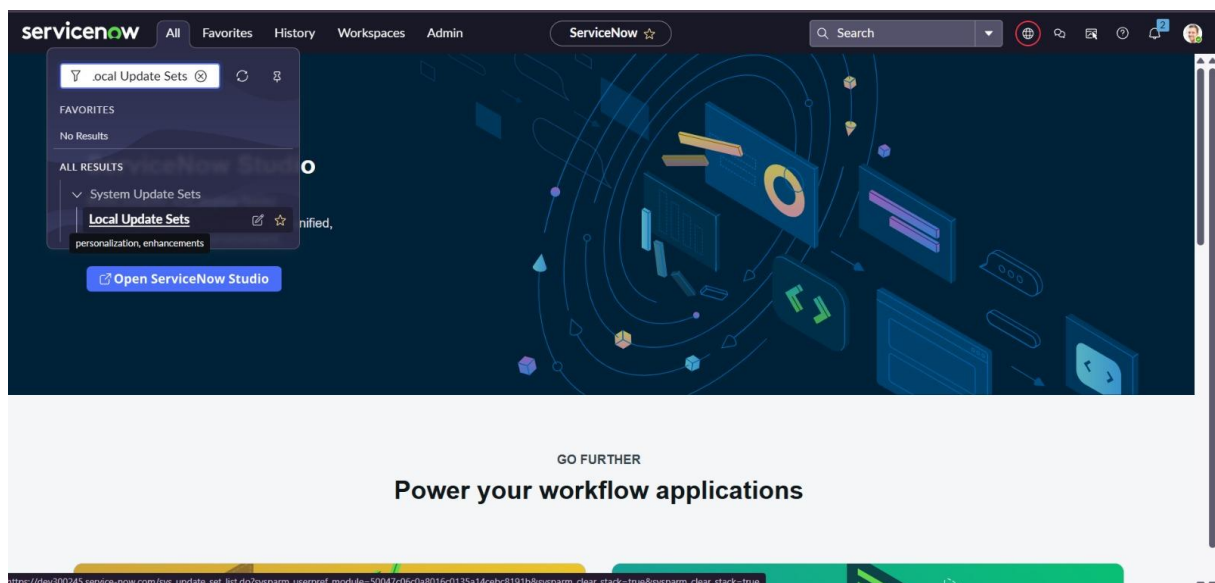
Go to Personal Developer Instance → Request Instance and fill in the required details.

Once your instance is ready, log in using the provided credentials to access ServiceNow.



## 2. Creation of New Update Set

In the filter navigator, search Local Update Set → click New.



Enter Name: Family Expenses, then click Submit and Make Current.

The screenshot shows the 'Update Set' form for 'Family Expenses'. The form is divided into two columns. The left column contains fields for: Name (Family Expenses), State (In progress), Parent (empty), Release date (empty), Install date (empty), Installed from (empty), and Description (empty). The right column contains fields for: Application (Global), Created (2025-10-29 22:10:32), Created by (admin), and Merged to (empty). At the bottom left are 'Update' and 'Delete' buttons. At the top right are 'Update' and 'Delete' buttons, along with up and down arrows.

### 3. Creation of Family Expenses Table

In the filter navigator, search Tables → click New.

Enter Label: Family Expenses, Menu Name: Family Expenditure, then right-click the header and select Save.

The screenshot shows the 'Table' form for 'Family\_Expenses'. The form is divided into two columns. The left column contains fields for: Label (Family\_Expenses) and Name (u\_family\_expenses). The right column contains fields for: Application (Global) and Remote Table (empty). At the bottom left are 'Delete', 'Update', and 'Delete All Records' buttons, along with up and down arrows.

### 4. Creation of Columns (Fields)

Add new rows under **Columns** with the following details:

- **Number** – String
- **Date** – Date
- **Amount** – Integer
- **Expense Details** – String (Max length: 800)



Open the Number field → Advanced view.

Check Use dynamic default and select Get Next Padded Number, then click Update.

The screenshot shows the 'Dictionary Entry Number' Advanced view configuration page. The top navigation bar includes a back arrow, a menu icon, the title 'Dictionary Entry Number View: Advanced\*', and action buttons 'Delete Column', 'Update', and arrows. The main configuration area is divided into two columns. The left column contains fields for: \* Table (Family\_Expenses [u\_family\_expenses]), \* Type (String), \* Column label (Number), \* Column name (u\_number), and \* Max length (40). The right column contains: Application (Global), Active (checked), Function field (unchecked), Read only (unchecked), Mandatory (unchecked), and Display (unchecked). Below these is a blue informational bar: 'Alters the behavior of a field or functionality that depends on the field. [More Info](#)'. Under 'Attributes' is an empty text box. At the bottom, there are tabs for 'Choice List Specification', 'Calculated Value', and 'Default Value' (which is selected). Below the tabs is another blue informational bar: 'The **Default value** specifies what value the field has when first displayed.' Below this, 'Use dynamic default' is checked, and 'Dynamic default value' is set to 'Get Next Padded Number'.

Search Number Maintenance → click New.

Set Table: Family Expenses, Prefix: MFE, and click Submit.

The screenshot shows the 'Number MFE' configuration page. The top navigation bar includes a back arrow, a menu icon, the title 'Number MFE', and action buttons 'Update', 'Delete', and arrows. The main configuration area contains fields for: \* Table (Family\_Expenses), Prefix (MFE), \* Number (1,000), Application (Global), and Number of digits (7). At the bottom, there are 'Update' and 'Delete' buttons, followed by 'Related Links' and a 'Show Counter' link.

## 5. Configure the Form

Search Family Expenses and open it.

Click New, then right-click the header → Configure → Form Design.

Arrange fields as needed.

Set Number as *Read-only* and make Date and Amount *Mandatory*.

Click Save.

Family\_Expenses [u\_fam] Default view Form Design

Fields Field Types

Filter

Fields

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

Formatters

- Activities (filtered)
- Contextual Search Results
- Ratings

Family\_Expense [u\_family\_expenses]

1 Column

- Number
- Expense Details

2 Column

- Date
- Amount

## 6. Creation of Daily Expenses Table

Search Tables → click New.

Table Daily\_Expenses

Delete Update Delete All Records

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](#)

\* Label Daily\_Expenses

Application Global

\* Name u\_daily\_expenses

Remote Table

Enter Label: Daily Expenses, add to Menu: Family Expenditure, then right-click the header and select Save.

Table Daily\_Expenses

Delete Update Delete All Records

\* Label Daily\_Expenses

Application Global

\* Name u\_daily\_expenses

Remote Table

Columns Controls Application Access

Table Columns for text Search

1 to 11 of 11 New

Dictionary Entries

Column label	Type	Reference	Max length	Default value	Display
Updated by	String	(empty)	40		false
Sys ID	Sys ID (GUID)	(empty)	32		false
Updates	Integer	(empty)	40		false
Expense	Integer	(empty)	40		false
Family Member Name	Reference	User	32		false
Updated	Date/Time	(empty)	40		false
comments	String	(empty)	800		false
Number	String	(empty)	40	javascript:getNextObjNumberPadded();	false
Created by	String	(empty)	40		false
Date	Date	(empty)	40		false

Columns

Controls

Application Access

Table Columns

for text

Search

1 to 11 of 11

New

Dictionary Entries

Column label	Type	Reference	Max length	Default value	Display
Updated by	String	(empty)	40		false
Sys ID	Sys ID (GUID)	(empty)	32		false
Updates	Integer	(empty)	40		false
Expense	Integer	(empty)	40		false
Family Member Name	Reference	User	32		false
Updated	Date/Time	(empty)	40		false
comments	String	(empty)	800		false
Number	String	(empty)	40	javascript:getNextObj(NumberPadded());	false
Created by	String	(empty)	40		false
Date	Date	(empty)	40		false
Created	Date/Time	(empty)	40		false
Insert a new row...					

Delete

Update

Delete All Records

Access Controls (4)

Security Data Filters

Labels (1)

Database Indexes (2)

Table Subscription Configuration (1)

Decision Type

Search

1 to 4 of 4

Actions on selected rows...

Access Controls

## 7.Making Number Field an Auto-Number

Open the Number field → Advanced view.

Enable Use dynamic default and choose Get Next Padded Number, then click Update

Dictionary Entry

Number View: Advanced

Delete Column

Update

↑

↓

\* Table

Daily\_Expenses [u\_daily\_expenses]

\* Type

String

\* Column label

Number

\* Column name

u\_number

\* Max length

40

Attributes

Application

Global

Active

☒

Function field

☐

Read only

☒

Mandatory

☐

Display

☐

Reference Specification

Choice List Specification

Function Definition

Dependent Field

Calculated Value

Default Value

Use dynamic default

☒

Dynamic default value

Get Next Padded Number

Delete Column

Update

Search Number Maintenance → New.

Set Table: Daily Expenses, Prefix: DFE, and click Submit.

Number DFE

\* Table:

Prefix:

\* Number:

Application:

Number of digits:

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

Related Links  
[Show Counter](#)

## 8.Configure the Form

Search Daily Expenses and open it.

Click New, then right-click the header → Configure → Form Design.

Arrange fields as needed.

Set Number as *Read-only* and make Date and Family Member Name *Mandatory*.

Click Save.

Daily\_Expenses [u\_daily\_expenses] Default view 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] 2 Column

- Number
- Date
- Expense

1 Column

- Comments
- Member name

## 9.Creating Relationship Between Tables

Search Relationships → click New.

Set Name: Daily Expenses, Applies to Table: Family Expenses, Related Table: Daily Expenses, then click Save.

Relationship  
Daily\_Expenses

Name:

Advanced ☐

Application:

Applies to table:

Queries from table:

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

```

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);

```

Run Query Diagnostics Update Delete

## 10. Configuring Related List

Open Family Expenses → New → Configure > Related Lists → add Daily Expenses → Save.

Configuring related lists on Family\_Expenses form

Available: Attachments

Selected: Daily\_Expenses

View name:

Cancel Save

Related Links  
[Show versions](#)  
[Related list performance diagnostics](#)

## 11. Business Rule Creation

Navigate to All → Business Rules → New.

Name: Family Expenses BR

Table: Daily Expenses

Add query: required



**Business Rule**  
Family Expenses BR

A 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. [More Info](#)

Name: Family Expenses BR

Table: Daily\_Expenses [u\_daily\_expenses]

Application: Global

Active ☒

Advanced ☒

**When to run** | Actions | Advanced

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 --

**Advanced**

Condition:

Script: ☒ Turn on ECMAScript 2021 (ES12) mode

```

1 (function executeRule(current, previous /*null when async*/) {
2
3
4   var FamilyExpenses = new GlideRecord('u_family_expenses');
5
6   FamilyExpenses.addQuery('u_date', current.u_date);
7
8   FamilyExpenses.query();
9
10  if(FamilyExpenses.next())
11  {
12    {
13
14      FamilyExpenses.u_amount += current.u_expense;
15
16      FamilyExpenses.u_expense_details += ">" + current.u_comments + ":" + "Rs." + current.u_expense + "/-";
17
18      FamilyExpenses.update();
19    }
20  }
21  else
22
23

```

## 12. Configure Relationship

Go to All → Relationships and open Daily Expenses Relationship.

Set Applies to table: Family Expenses.

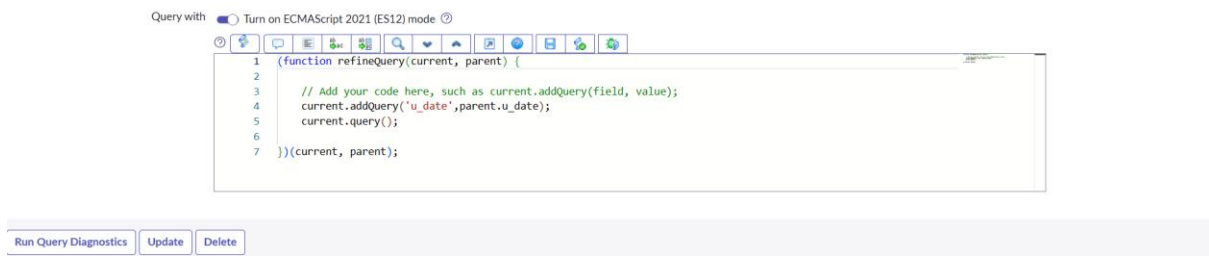
Add Query:

```

(function refine Query (current, parent) {
  current. add Query('update', parent. u_date);

  current. query ();
}) (current, parent);

```



Update the Relationships by clicking on update.

- Built in **ServiceNow** to manage family spending.
- **Daily Expenses** table records:
  - Date, Amount, Comments, Family Member
- **Family Expenses** table shows:
  - Total spent per date, with summary
- Tables are **linked**:
  - Daily entries update family totals automatically
- Helps track spending and keep financial records organized.

Daily_Expenses					
Number		Search		Actions on selected rows... New	
All					
Number	comments	Date	Expense	Family Member Name	
DFE0001003	money	2025-10-31	500	Abraham Lincoln	
DFE0001002	Mobile	2025-11-01	500	Abel Tuter	

Family_Expenses				
Number		Search		Actions on selected rows... New
All				
Number	Amount	Date	Expense Details	
MFE0001011	1,000	2025-10-31	new	
MFE0001009	500	2025-11-01	Mobile	

Family\_Expensess

MFE0001009

Update

Delete

Number

MFE0001009

Expense Details

Mobile

Date

2025-11-01

Amount

500

Update

Delete

Daily\_Expensess

Number

Search

Actions on selected rows...

New

Daily\_Expensess

	Number	comments	Date	Expense	Family Member Name
<input type="checkbox"/>	DFE0001002	Mobile	2025-11-01	500	Abel Tuter

1 to 1 of 1

## Conclusion:

In conclusion, the *Family Expenses Calculation System* built on ServiceNow provides an efficient and organized way to manage household finances. By leveraging ServiceNow's automation and data management capabilities, the system simplifies expense tracking, ensures accuracy, and offers real-time insights into family spending patterns. This project not only enhances financial transparency but also promotes better budgeting and informed decision-making—ultimately contributing to improved financial stability and well-being for families.

Done By,

**Team ID:**

Team Member: H.Naveen Sam Raj

Team Member: P.Selva Bharath

Team Member: E.Vasanthan

Team Member: T.Sirajudeen

# Thank You!