

## **Calculating Family Expenses Using ServiceNow**

**Team Id: NM2025TMID15476**

### **Team Members:**

**Team Leader: SATHYA KALA K**

**Team Member 1: ASWINI B**

**Team Member 2: JEMIMA A**

**Team Member 3: MEENA M**

### **Problem Statement:**

Managing daily family expenses is often done in notebooks or mobile notes, which is not well-organized and can easily be forgotten. Without a proper system, it is hard to keep track of how much money is spent and where it is spent. A simple automated solution is needed to handle this problem.

### **Objective:**

- To design a ServiceNow-based system for managing family spending.
- To provide a central place to record all daily expenses.
- To use automation for updating totals and maintaining records.
- To improve accuracy and reduce manual work in expense calculation.
- To support future improvements like reports and dashboards.

**Skills:**

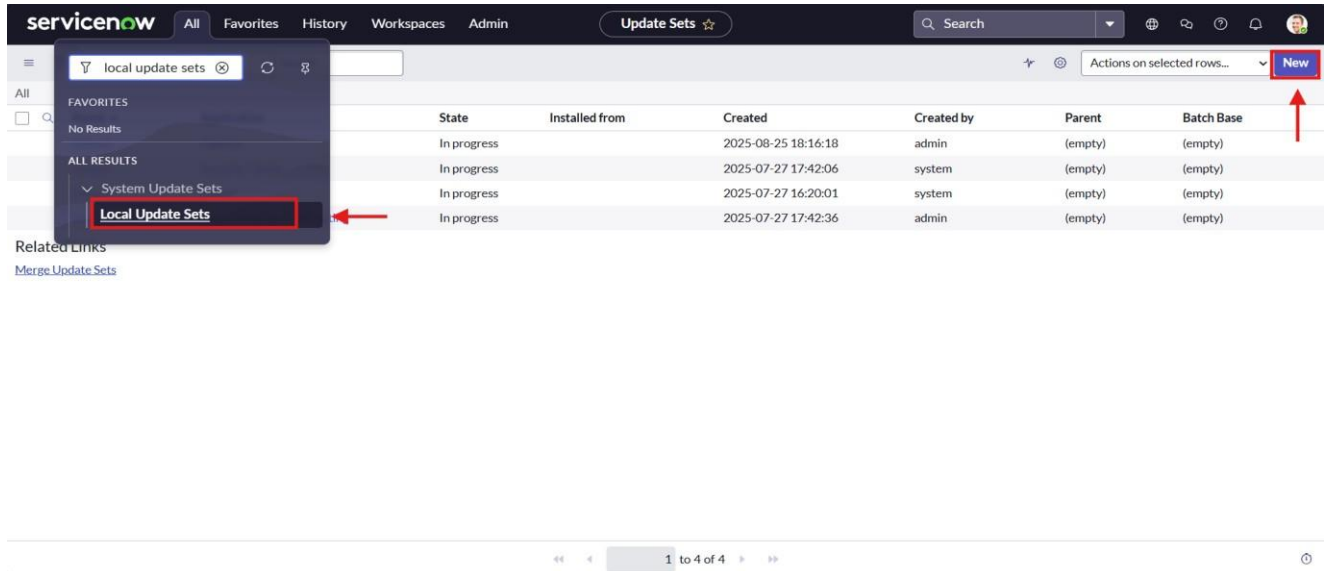
- ServiceNow basics
- Tables and Relationships
- Update Sets
- Business Rules
- Flow Designer

**TASK INITIATION****Milestone 1 : Setting Up ServiceNow Instance****Activity 1: Create Instance**

1. Go to [developer.servicenow.com](https://developer.servicenow.com).
2. Sign up for a free developer account.
3. After logging in, click **Personal Developer Instance**.
4. Click **Request Instance** to create your own ServiceNow space.
5. Fill in the form and submit it.
6. You will get an email with your instance link, username, and password.
7. Log in with those details.
8. Now your ServiceNow instance is ready to use.

**Milestone 2 : Creation of New Update Set****Activity 1: Create Update Set**

1. Go to **All** in ServiceNow.
2. In the filter search box, type **Local Update Sets** and click on it.
3. Click on **New**.

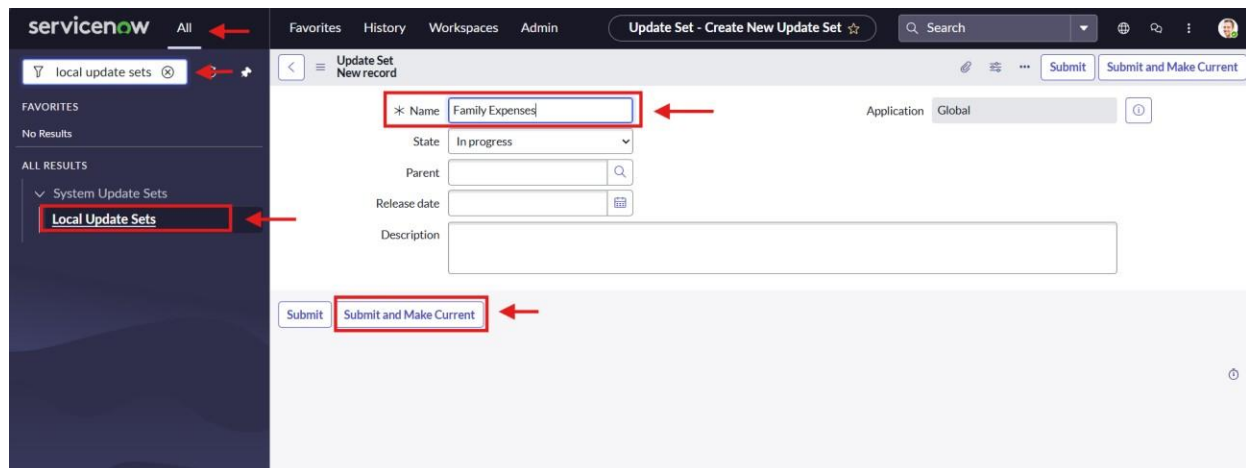


State	Installed from	Created	Created by	Parent	Batch Base
In progress		2025-08-25 18:16:18	admin	(empty)	(empty)
In progress		2025-07-27 17:42:06	system	(empty)	(empty)
In progress		2025-07-27 16:20:01	system	(empty)	(empty)
In progress		2025-07-27 17:42:36	admin	(empty)	(empty)

4. Enter the details:

- **Name:** Family Expenses

5. Click on **Submit and Make Current**.



Update Set - Create New Update Set

\* Name:

State:

Parent:

Release date:

Description:

Application:

## Milestone 3 : Creation of Table

### Activity 1: Create Family Expenses Table

1. In ServiceNow, go to **All**.
2. In the filter search box, type **Tables**.
3. Click on **Tables** under *System Definition*.
4. Click on **New**.

servicenow All

Favorites History Workspaces Admin

Tables ☆

Search

Actions on selected rows... New

All > Update name is not empty

<input type="checkbox"/>	Label	Name	Extends table	Extensible	Updated
<input type="checkbox"/>	Account Subscription Entitlement	account_subscription_entitlement	(empty)	false	2025-07-27 16:55:49
<input type="checkbox"/>	Adaptive Authentication Event	adaptive_auth_event	(empty)	false	2025-07-27 16:37:47
<input type="checkbox"/>	Agent Assist Recommendation	agent_assist_recommendation	Application File	false	2025-07-27 16:39:11
<input type="checkbox"/>	MID Server File	agent_file	(empty)	false	2025-07-27 16:23:06
<input type="checkbox"/>	Record Producer Configuration	aisa_rp_config	Application File	false	2025-07-27 16:42:56
<input type="checkbox"/>	Search Actions	aisa_ui_action	Application File	false	2025-07-27 16:42:56
<input type="checkbox"/>	AI Search ACL Overrides	ais_acl_overrides	Application File	false	2025-07-27 16:06:27
<input type="checkbox"/>	AI Search Active Table Ingestion Tracker	ais_active_table_ingestion_tracker	(empty)	false	2025-07-27 16:06:25
<input type="checkbox"/>	AI Search Async Genius Result	ais_async_genius_result	(empty)	false	2025-07-27 16:06:29
<input type="checkbox"/>	AI Search Async Request	ais_async_request	(empty)	false	2025-07-27 16:06:29
<input type="checkbox"/>	AI Search Child Table	ais_child_table	Application File	false	2025-07-27 16:06:28
<input type="checkbox"/>	AI Search Configuration Attribute	ais_configuration_attribute	(empty)	false	2025-07-27 16:06:25
<input type="checkbox"/>	AI Search Connection	ais_connection	(empty)	false	2025-07-27 16:06:29
<input type="checkbox"/>	AI Search Country To Search Language	ais_country_to_search_language	Application File	false	2025-07-27 16:06:29
<input type="checkbox"/>	Custom Matcher	ais_custom_matcher	Application File	false	2025-07-27 16:06:27
<input type="checkbox"/>	AI Search Indexed Source	ais_datasource	Application File	false	2025-07-27 16:06:28

1 to 20 of 5,633

5. Enter the details:

- **Label:** Family Expenses
- **Name:** (Auto-populated)
- **New menu name:** Family Expenditure

6. Go to the header, right-click, and select **Save**.

Favorites History Workspaces Admin

Table - New Record ☆

Search

Submit Cancel

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 Family Expenses

\* Name u\_family\_expenses

Extends table

Application Global

Create module ☒

Create mobile module ☒

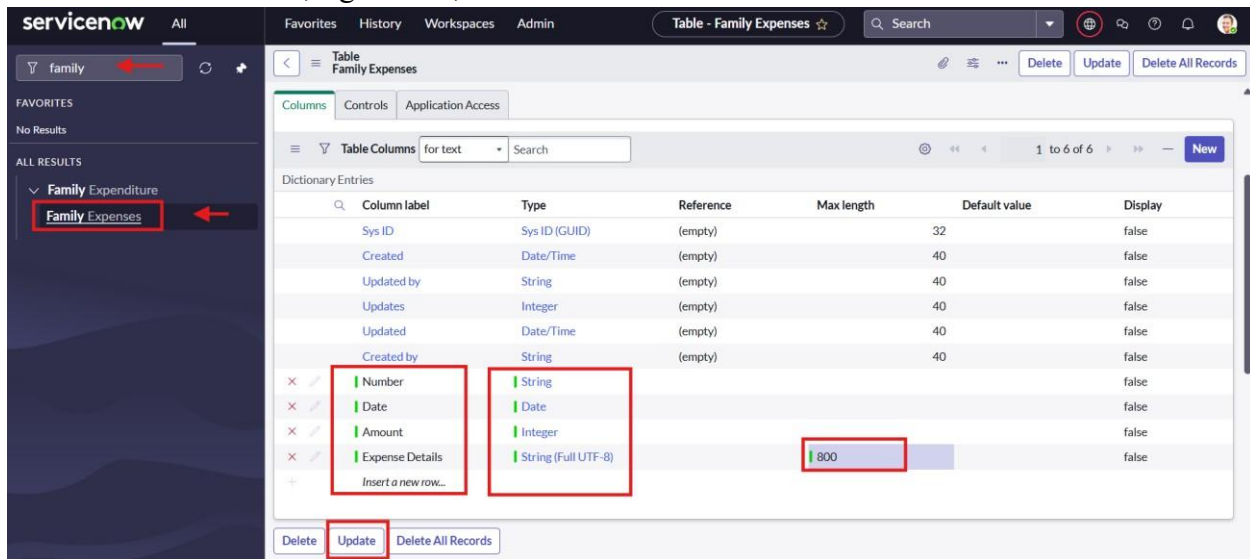
Add module to menu -- Create new --

New menu name Family Expenditure

Remote Table ☐

## Activity 2: Create Table

1. In the **Family Expenses** table, go to the **Columns** section.
2. Double-click near **Insert a new row**.
3. Enter the details:
  - **Column label:** Number
  - **Type:** String
  - Again double-click.
  - **Column label:** Date
  - **Type:** Date
  - Again double-click.
  - **Column label:** Amount
  - **Type:** Integer
  - Again double-click.
  - **Column label:** Expense Details
  - **Type:** String
  - **Max length:** 800
4. Go to the header, right-click, and select **Save**.



The screenshot shows the ServiceNow interface for configuring the 'Family Expenses' table. The 'Columns' tab is selected, and a new row is being added at the bottom. The columns being added are:

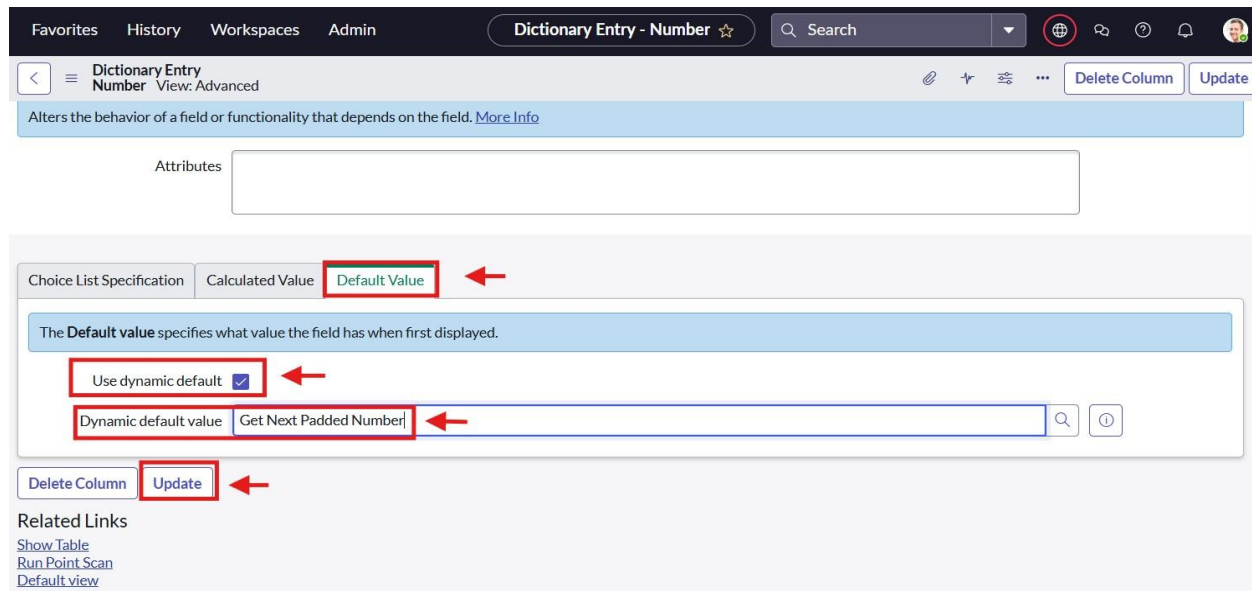
Column label	Type	Reference	Max length	Default value	Display
Sys ID	Sys ID (GUID)	(empty)	32		false
Created	Date/Time	(empty)	40		false
Updated by	String	(empty)	40		false
Updates	Integer	(empty)	40		false
Updated	Date/Time	(empty)	40		false
Created by	String	(empty)	40		false
Number	String				false
Date	Date				false
Amount	Integer				false
Expense Details	String (Full UTF-8)		800		false

The 'Update' button is highlighted at the bottom of the configuration page.

## Activity 3: Configure Auto Number for Family Expenses Table

1. Open the **Family Expenses** table.
2. Double-click the **Number** field/column.

3. Scroll down and switch to **Advanced view**.
4. In **Default Value**:
  - Tick **Use dynamic default**.
  - Select **Get Next Padded Number**.
5. Click **Update**.



Dictionary Entry - Number ☆

Search

Dictionary Entry Number View: Advanced

Alters the behavior of a field or functionality that depends on the field. [More Info](#)

Attributes

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**

Related Links

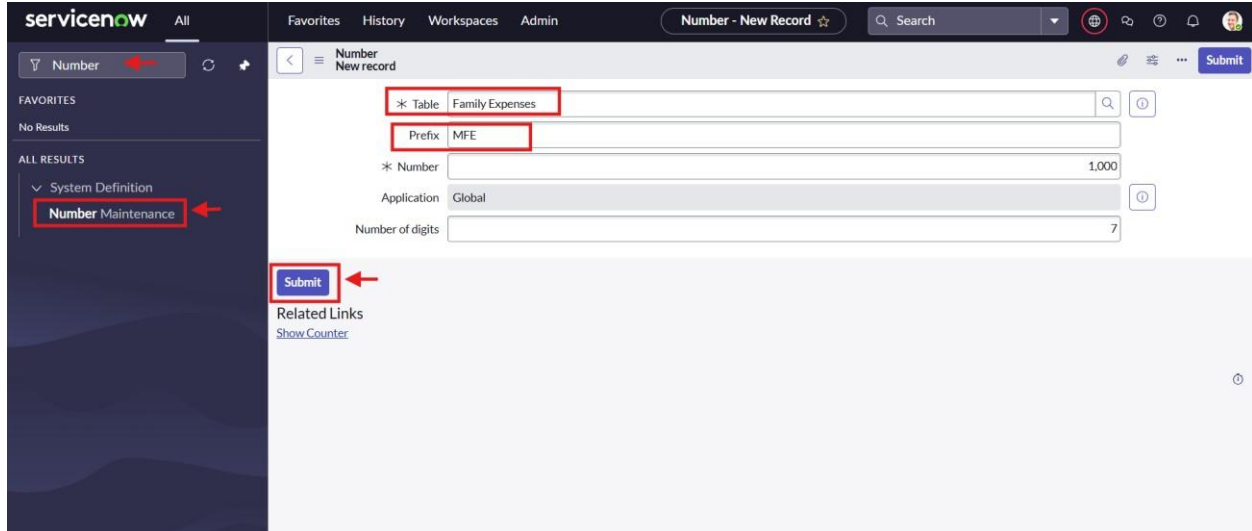
[Show Table](#)

[Run Point Scan](#)

[Default view](#)

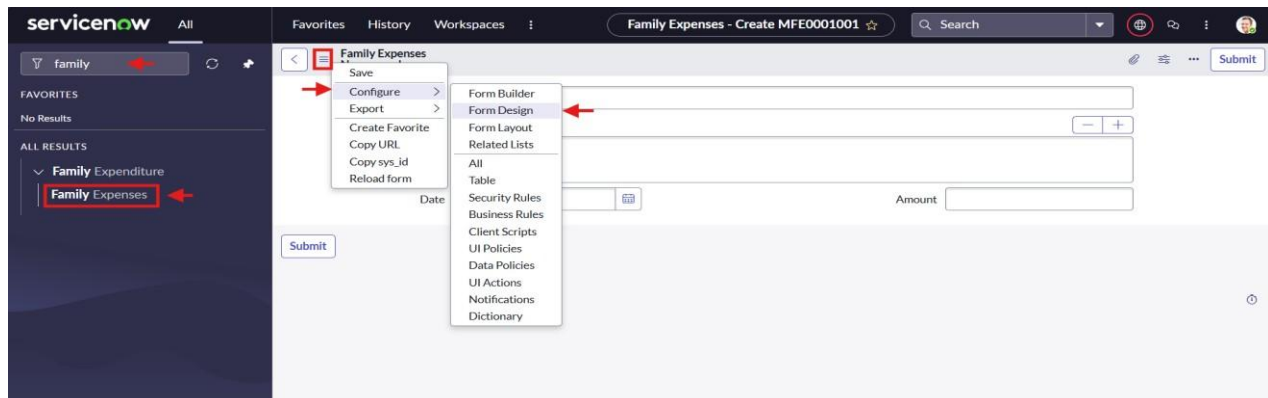
## Activity 4: Create Number Maintenance

1. In ServiceNow, go to **All**.
2. In the search box, type **Number Maintenance** and select it.
3. Click **New**.
4. Enter the details:
  - **Table:** Family Expenses
  - **Prefix:** MFE
5. Click **Submit**.

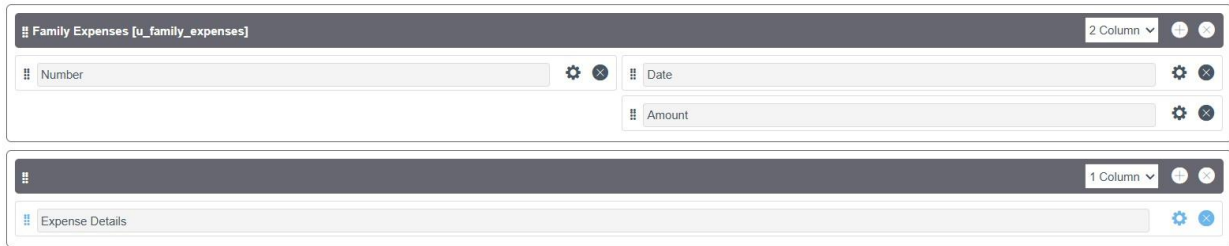


## Activity 5: Configure the Form

1. In ServiceNow, go to **All** and search for **Family Expenses**.
2. Open **Family Expenses** and click on **New**.
3. In the header, right-click → select **Configure** → **Form Design**.



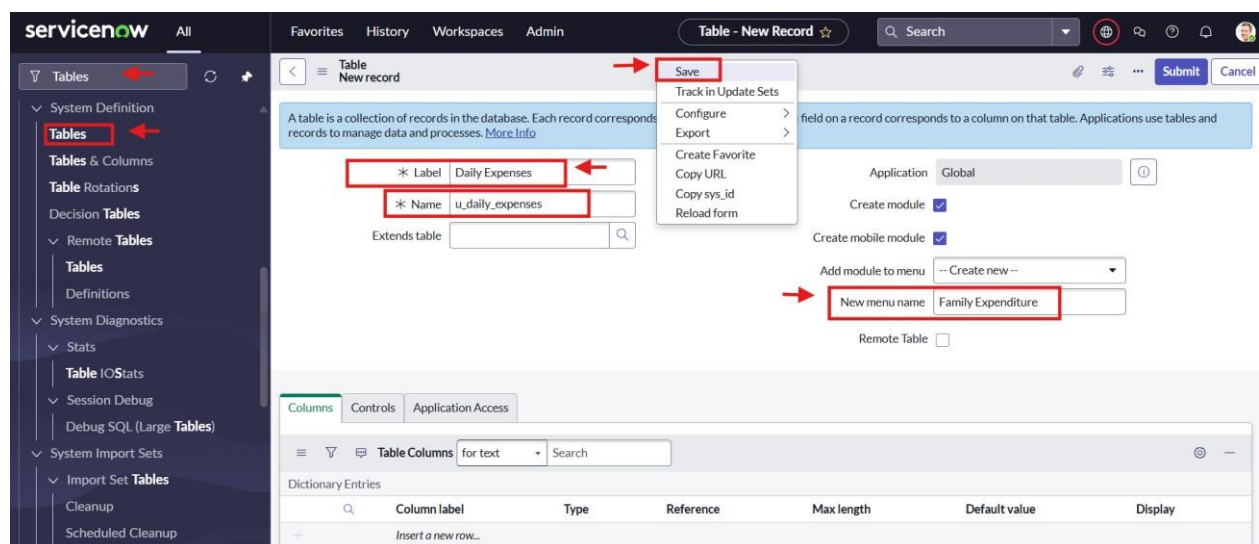
4. Arrange or drag & drop fields as needed.
5. For **Number** field: set it to **Read-Only** (using the gear icon).
6. For **Date** and **Amount** fields: set them as **Mandatory** (using the gear icon).
7. Click **Save**.



## Milestone 4 : Creation of Daily Expenses Table

### Activity 1: Create Daily Expenses Table

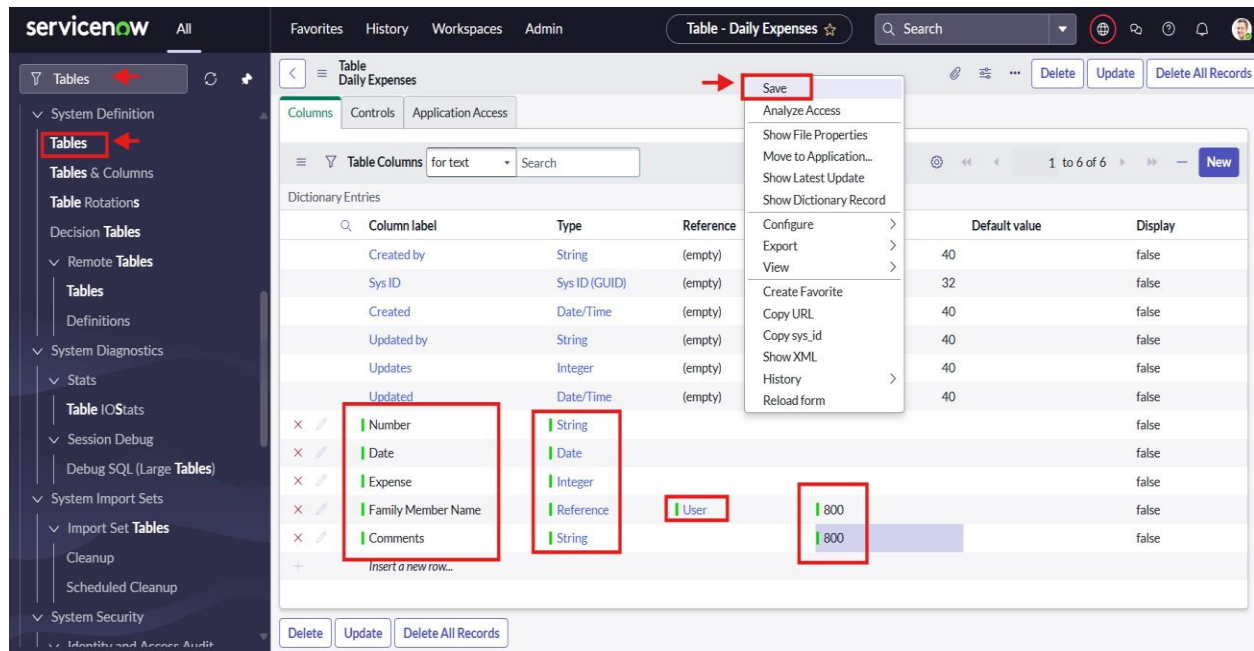
1. In ServiceNow, go to **All**.
2. In the search bar, type **Tables** and click on it.
3. Click on **New**.
4. Enter the details:
  - **Label:** Daily Expenses
  - **Name:** (Auto-populated)
  - **Add Module to menu:** Family Expenditure
5. In the header, right-click and click **Save**.





## Activity 2: Add Columns (Fields) to Daily Expenses Table

1. In the **Daily Expenses** table, go to the **Columns** section.
2. Double-click on **Insert a new row** to add fields.
3. Add the following columns one by one:
  - **Column label:** Number → **Type:** String
  - **Column label:** Date → **Type:** Date
  - **Column label:** Expense → **Type:** Integer
  - **Column label:** Family Member Name → **Type:** Reference (Max length: 800) **Reference:** User there show sys\_user
  - **Column label:** Comments → **Type:** String (Max length: 800)
4. After adding all columns, go to the header, right-click, and click **Save**.



Column label	Type	Reference
Created by	String	(empty)
Sys ID	Sys ID (GUID)	(empty)
Created	Date/Time	(empty)
Updated by	String	(empty)
Updates	Integer	(empty)
Updated	Date/Time	(empty)
Number	String	
Date	Date	
Expense	Integer	
Family Member Name	Reference	User
Comments	String	800

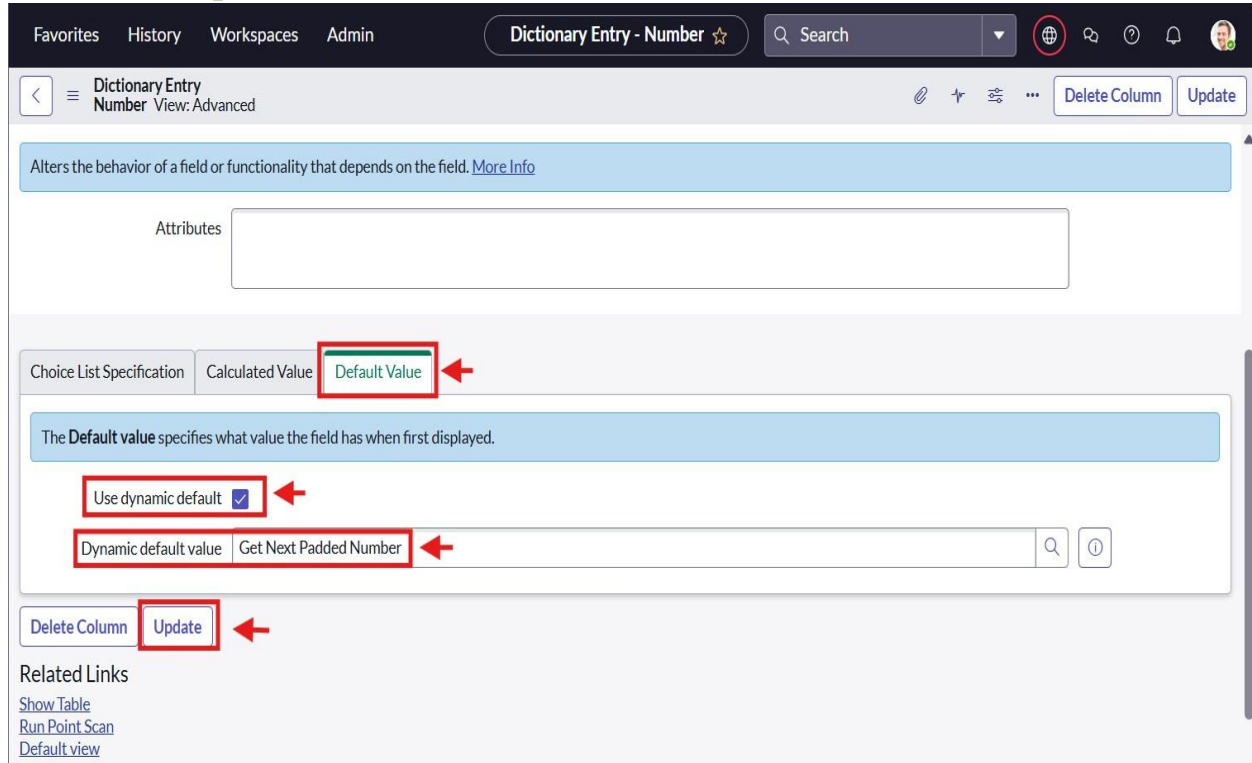
## Activity 3: Make Number Field Auto-Generate

1. In the **Daily Expenses** table, double-click on the **Number** field.
2. Scroll down and switch to **Advanced view**.

### 3. In **Default Value**:

- Tick **Use dynamic default**.
- Select **Get Next Padded Number**.

### 4. Click **Update**.



Dictionary Entry - Number ☆ Search

Dictionary Entry Number View: Advanced Delete Column Update

Alters the behavior of a field or functionality that depends on the field. [More Info](#)

Attributes

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**

Related Links

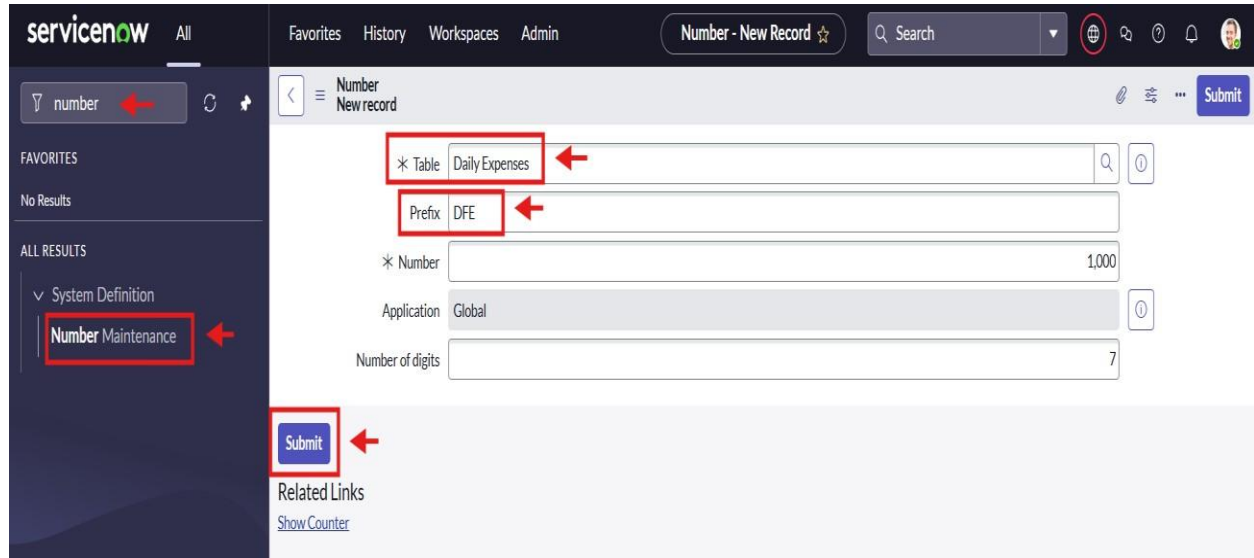
[Show Table](#)

[Run Point Scan](#)

[Default view](#)

## Activity 4: Number Maintenance

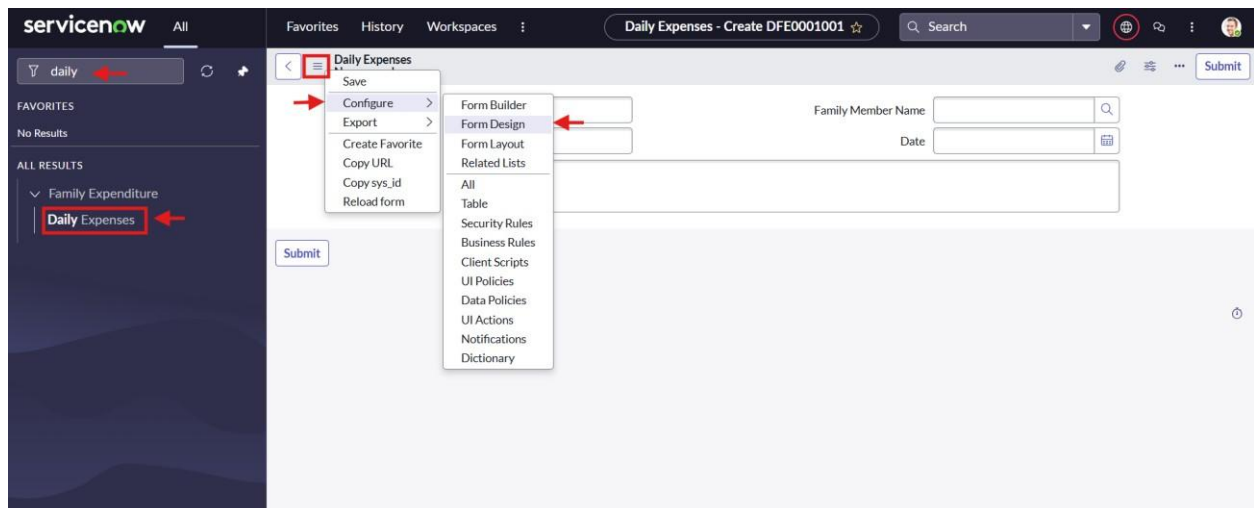
1. In ServiceNow, go to **All**.
2. Search for **Number Maintenance** and open it.
3. Click on **New**.
4. Enter the details:
  - **Table:** Daily Expenses •
  - Prefix:** DFE (Daily Expenses)
5. Click **Submit**.



ServiceNow interface showing the 'Number - New Record' form. The left sidebar highlights 'Number Maintenance' under 'System Definition'. The main form fields include: Table (Daily Expenses), Prefix (DFE), Number (1,000), Application (Global), and Number of digits (7). A 'Submit' button is visible at the bottom left.

## Activity 5: Configure the Form

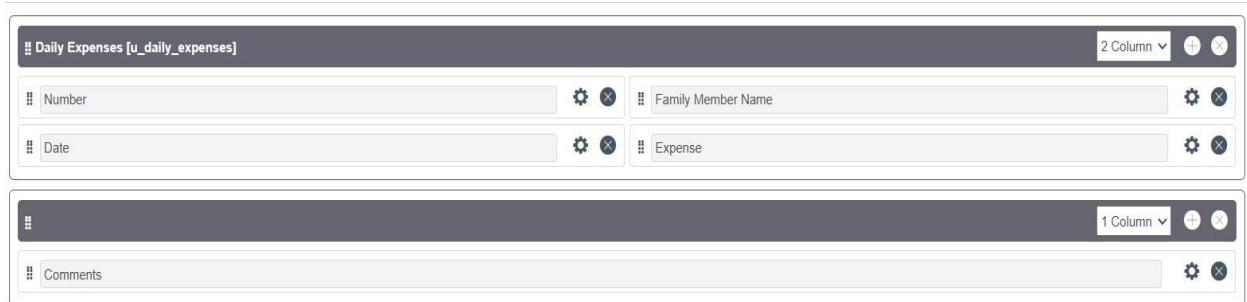
1. In ServiceNow, go to **All** and search for **Daily Expenses**.
2. Open **Daily Expenses** and click on **New**.
3. In the header, right-click → choose **Configure** → **Form Design**.



ServiceNow interface showing the 'Daily Expenses - Create DFE0001001' form. The left sidebar highlights 'Daily Expenses' under 'Family Expenditure'. The main form fields include: Family Member Name and Date. A context menu is open over the 'Daily Expenses' header, showing options like 'Form Builder', 'Form Design', 'Form Layout', and 'Related Lists'.

4. Arrange or drag & drop the fields as needed.
5. For the **Number** field: set it to **Read-Only** (using the gear icon).

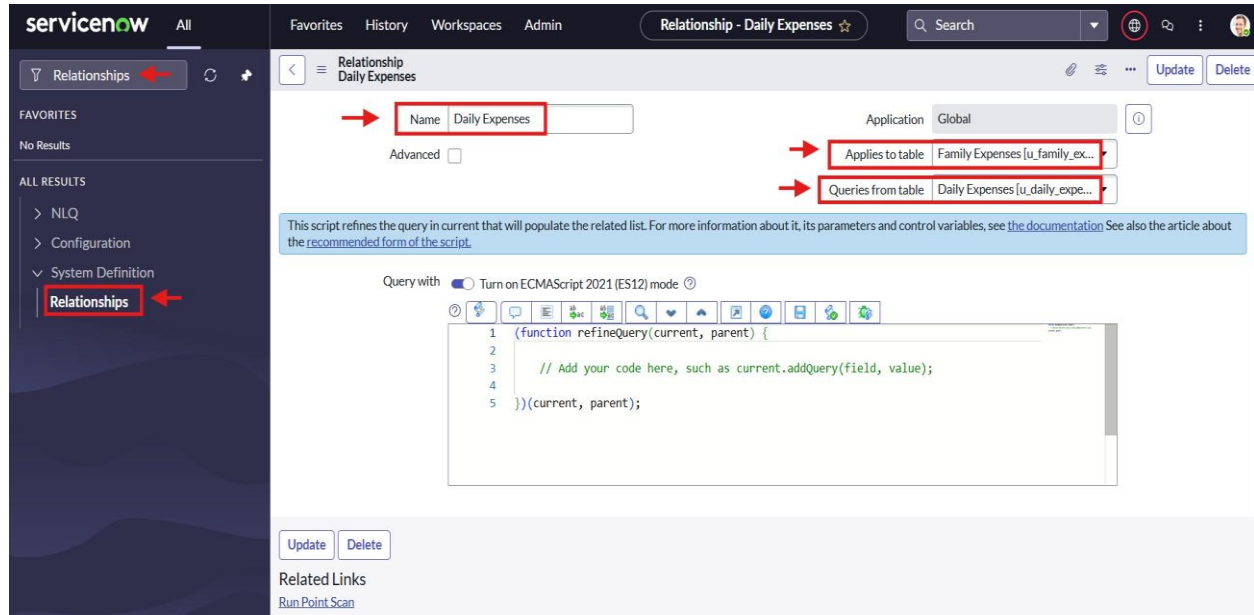
6. For **Date** and **Family Member Name** fields: set them as **Mandatory** (using the gear icon).
7. Click **Save**.



## Milestone 5 : Creation of Relationship between Tables

### Activity 1: Link Family Expenses and Daily Expenses Tables

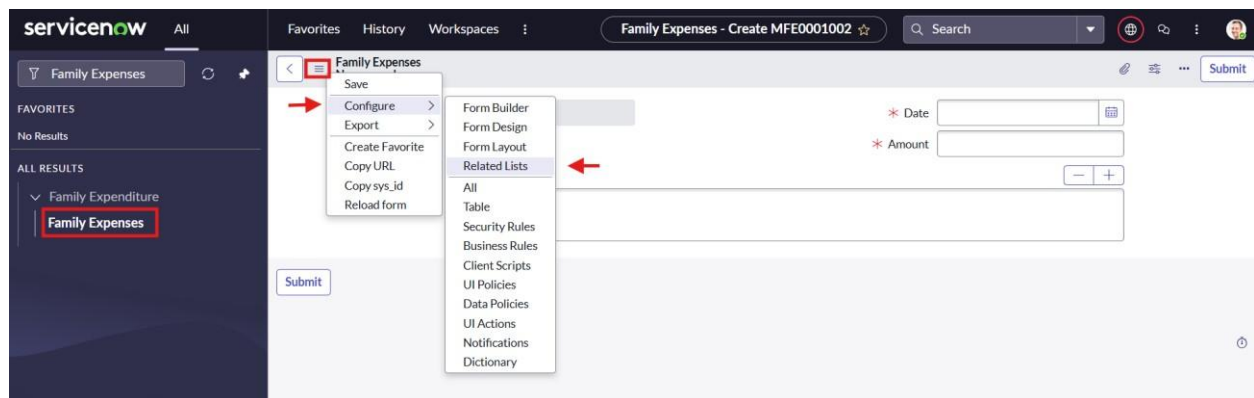
1. In ServiceNow, go to **All**.
2. In the search box, type **Relationships** and open it.
3. Click on **New**.
4. Enter the details:
  - **Name:** Daily Expenses
  - **Applies to table:** Family Expenses
  - **Related table:** Daily Expenses
5. Click **Save**.



## Milestone 6 : Configuring Related List on Family Expenses

### Activity 1: Add Daily Expenses as Related List

1. In ServiceNow, go to **All** and search for **Family Expenses**.
2. Open **Family Expenses** and click on **New**.
3. In the header, right-click → choose **Configure** → **Related Lists**.



4. From the options, add **Daily Expenses** to the selected area.
5. Click **Save**.

< Configuring related lists on Family Expenses form

Available

- Attachments
- Daily Expenses**

>

<

Selected

- None--

>

<

Cancel Save

Available

- Attachments

>

<

Selected

- Daily Expenses**

>

<

Cancel **Save**

## Milestone 7 : Creation of Business Rules

### Activity 1: Add Business Rule for Family Expenses

1. In ServiceNow, go to **All**.
2. In the search bar, type **Business Rules** and select it under *System Definition*.
3. Click on **New**.
4. Enter the details:
  - **Name:** Family Expenses BR
  - **Table:** Daily Expenses
  - Check **Advanced**.

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\_expense]

Application: Global  
 Active: ☒  
 Advanced: ☒

5. Under When to run, check Insert and Update.

Business Rule Family Expenses BR

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 --  
 Role conditions: [Edit](#)

6. In the Advanced section, paste the following script:

```
(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 {
```

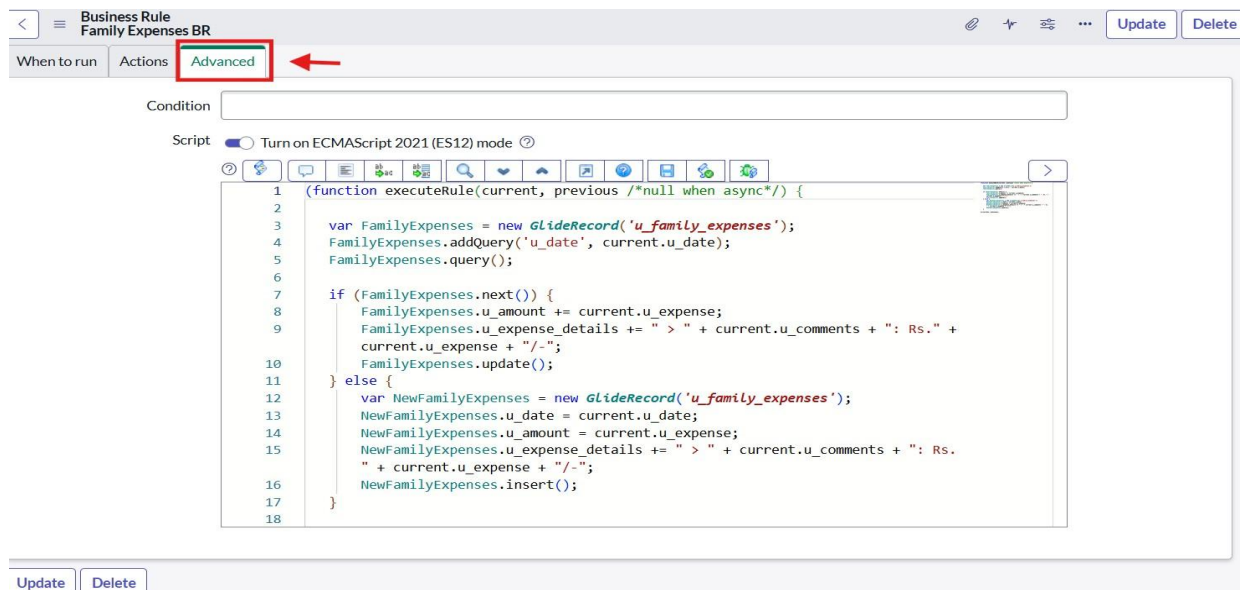
```

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);

```

7. Go to the header, right-click, and click Save.



## Milestone 8: Flow

### Activity 1: Update Daily Expenses Relationship

1. In ServiceNow, go to **All**.
2. In the search box, type **Relationships** and open it.
3. Open the **Daily Expenses Relationship**.
4. For **Applies to table**, select **Family Expenses**.
5. In **Query with**, paste the following script:

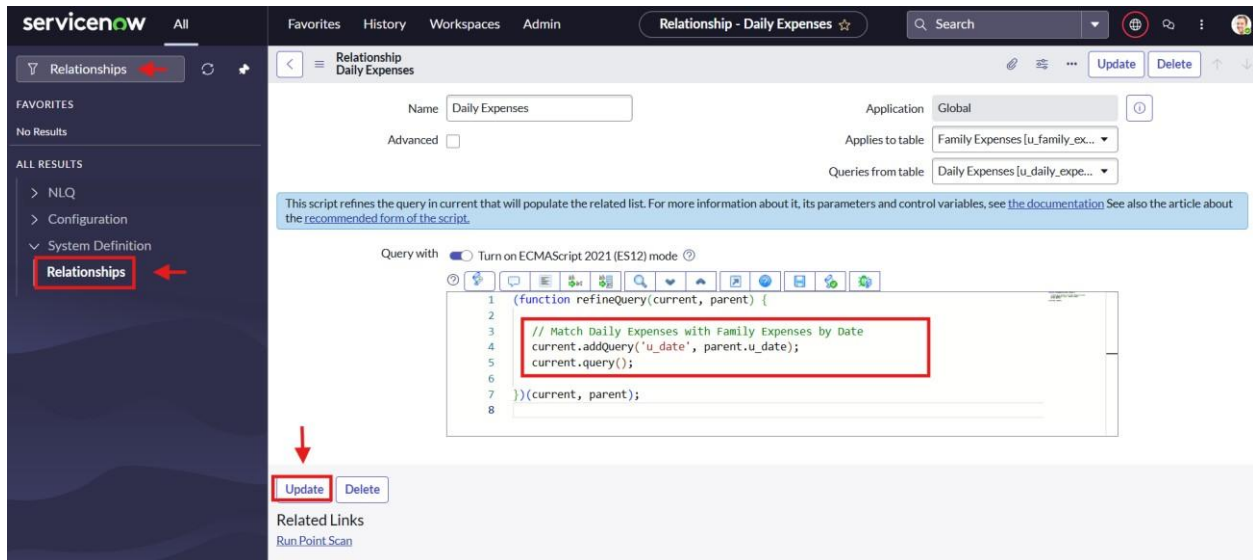


```
(function refineQuery(current, parent) {
```

```
    // Match Daily Expenses with Family Expenses by Date
    current.addQuery('u_date', parent.u_date);
    current.query();
```

```
})(current, parent);
```

6. Click **Update**.



The screenshot shows the ServiceNow interface for configuring a relationship. The left sidebar has a 'Relationships' menu item highlighted with a red box and an arrow. The main content area is titled 'Relationship - Daily Expenses'. It includes fields for 'Name' (Daily Expenses), 'Application' (Global), 'Applies to table' (Family Expenses [u\_family\_ex...]), and 'Queries from table' (Daily Expenses [u\_daily\_expe...]). A script editor is open, showing the following JavaScript code:

```

1 (function refineQuery(current, parent) {
2
3     // Match Daily Expenses with Family Expenses by Date
4     current.addQuery('u_date', parent.u_date);
5     current.query();
6
7 })(current, parent);
8

```

The 'Update' button at the bottom is highlighted with a red box and an arrow.

## **Conclusion :**

The project “**Calculating Family Expenses using ServiceNow**” helps in recording and managing daily spending in an easy way.

By using tables, relationships, and business rules, the system automatically updates and organizes expenses.

It makes the process faster and reduces errors compared to writing in books or notes. This project shows how ServiceNow can be used for simple family applications, not only for business use.

In the future, it can be improved by adding expense reports, charts, and mobile-friendly access.