

Calculating Family Expenses Using ServiceNow

Team Id: NM2025TMID12805

Team Members:

Team Leader: Rithika T

Team Member 1: Roshini S

Team Member 2: Sandwana krishnan N

Team Member 3: Aditi jain

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:

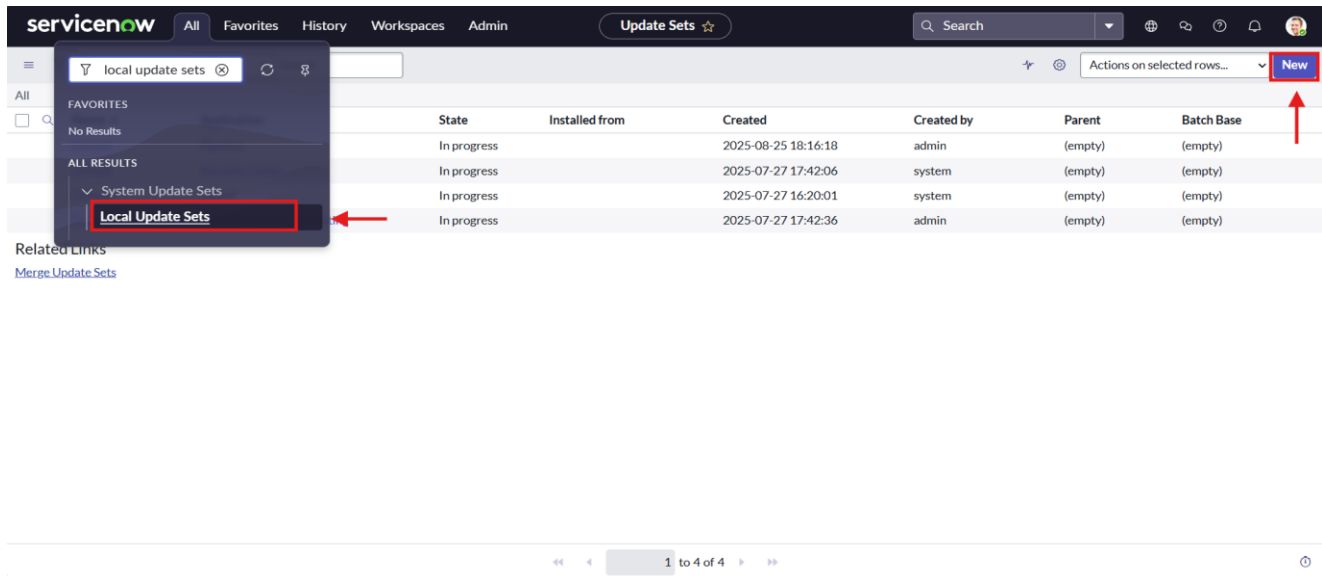
- ServiceNow basics
- Tables and Relationships
- Update Sets
- Business Rules
- Flow Designer

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

1. Go to developer.servicenow.com.
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**.



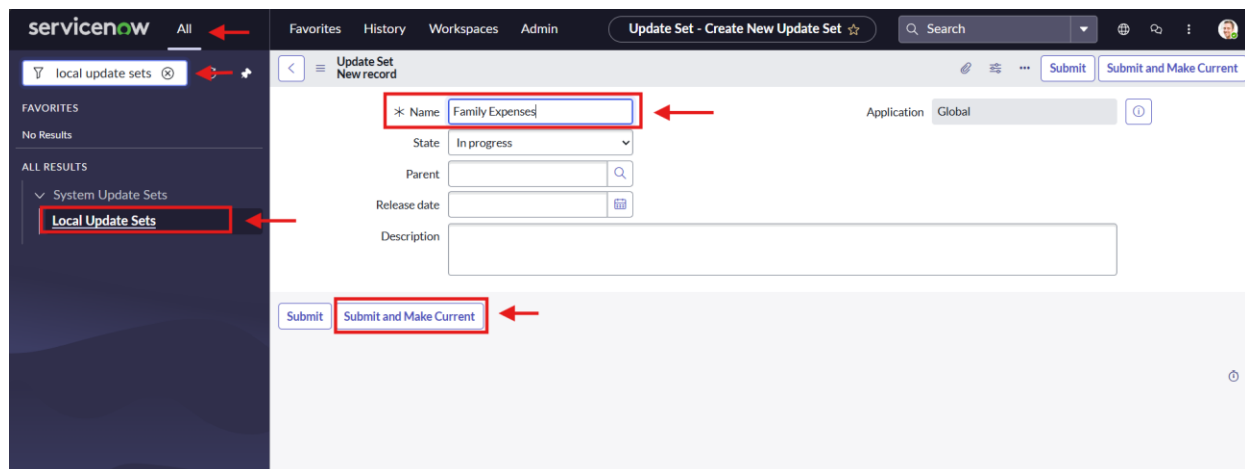
The screenshot shows the ServiceNow 'Update Sets' list. A dropdown menu is open for 'local update sets', and the 'Local Update Sets' option is selected. A table of update sets is displayed with columns: State, Installed from, Created, Created by, Parent, and Batch Base. A 'New' button is visible in the top right corner.

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**.



The screenshot shows the 'Update Set - Create New Update Set' form. The 'Name' field is filled with 'Family Expenses'. The 'State' dropdown is set to 'In progress'. The 'Application' dropdown is set to 'Global'. The 'Submit and Make Current' button is highlighted.

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** Actions on selected rows... **New**

All > Update name is not empty

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

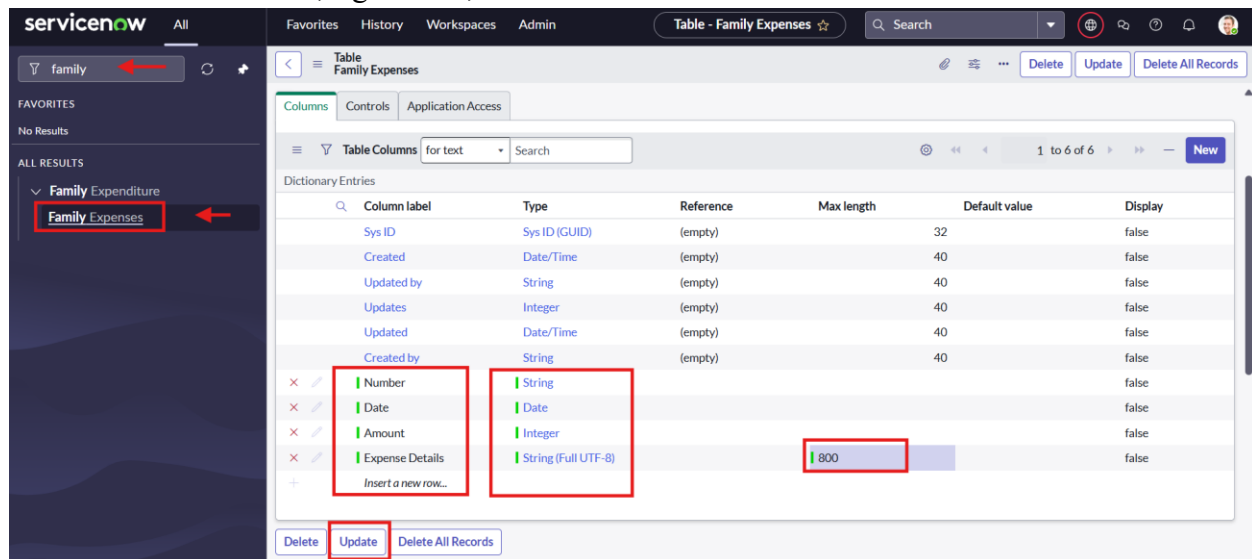
Table - New Record **Submit** **Cancel**

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	Application	Global
* Name	u_family_expenses	Create module	<input checked="" type="checkbox"/>
Extends table	<input type="text"/>	Create mobile module	<input checked="" type="checkbox"/>
		Add module to menu	-- Create new --
		New menu name	Family Expenditure
		Remote Table	<input type="checkbox"/>

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 'Table - Family Expenses' configuration page in the ServiceNow interface. The 'Columns' tab is active, and a new row is being added at the bottom. The new row has the following details:

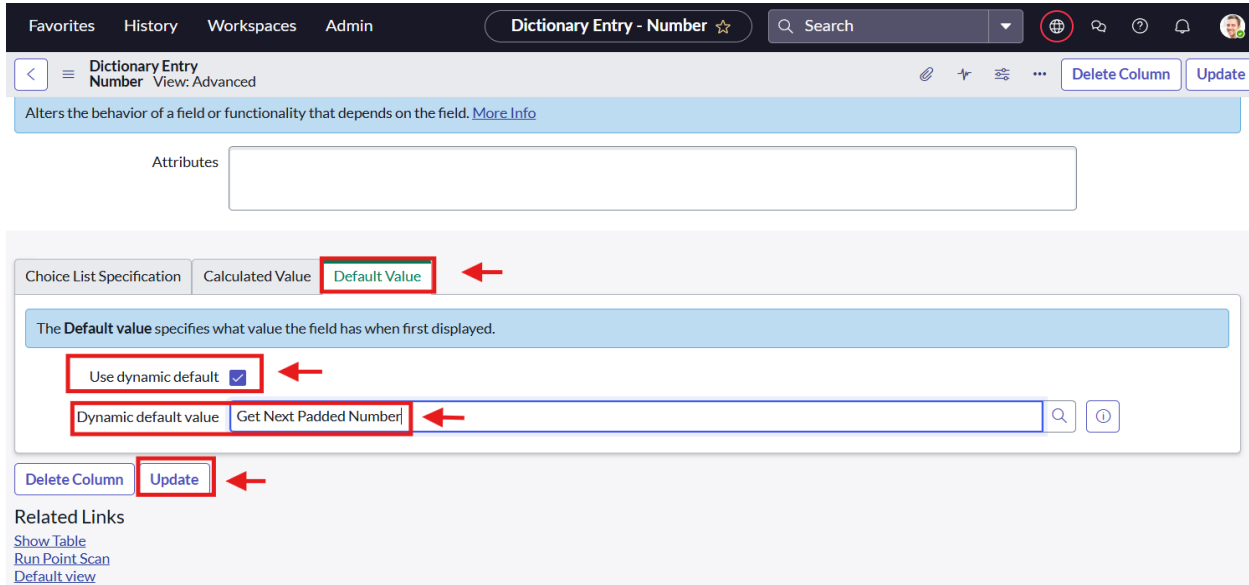
Column label	Type	Reference	Max length	Default value	Display
Sys ID	Sys ID (GUID)	(empty)	32		false
Created	Date/Time	(empty)	40		false
Updated by	String	(empty)	40		false
Updates	Integer	(empty)	40		false
Updated	Date/Time	(empty)	40		false
Created by	String	(empty)	40		false
Number	String		800		false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

The 'Update' button is highlighted with a red box.

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**.



Favorites History Workspaces Admin 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 ☒

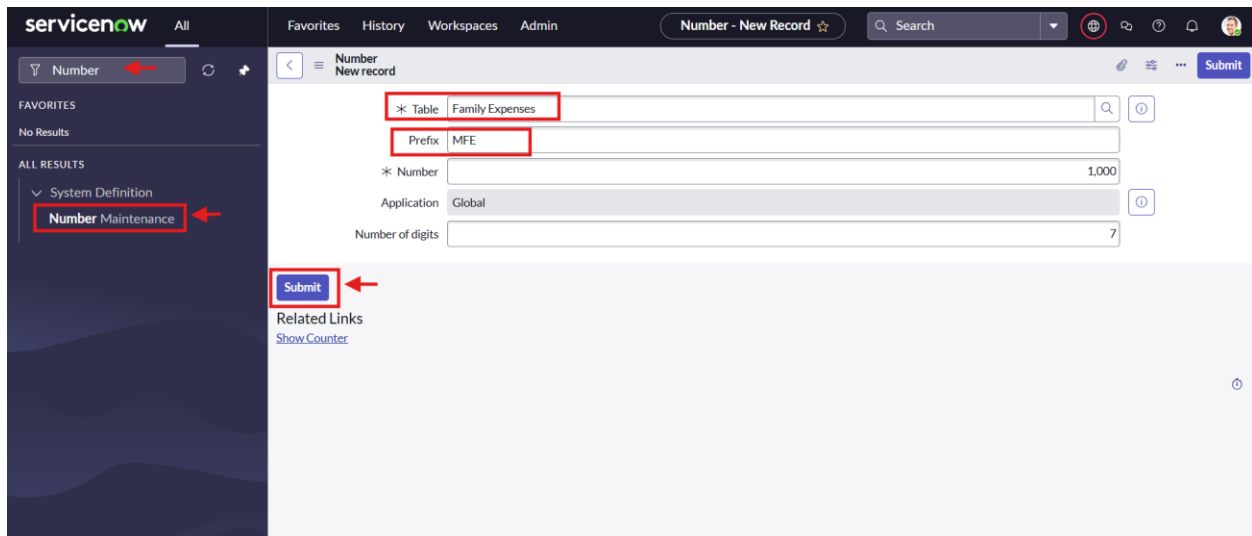
Dynamic default value Get Next Padded Number

Delete Column **Update**

Related Links
[Show Table](#)
[Run Point Scan](#)
[Default view](#)

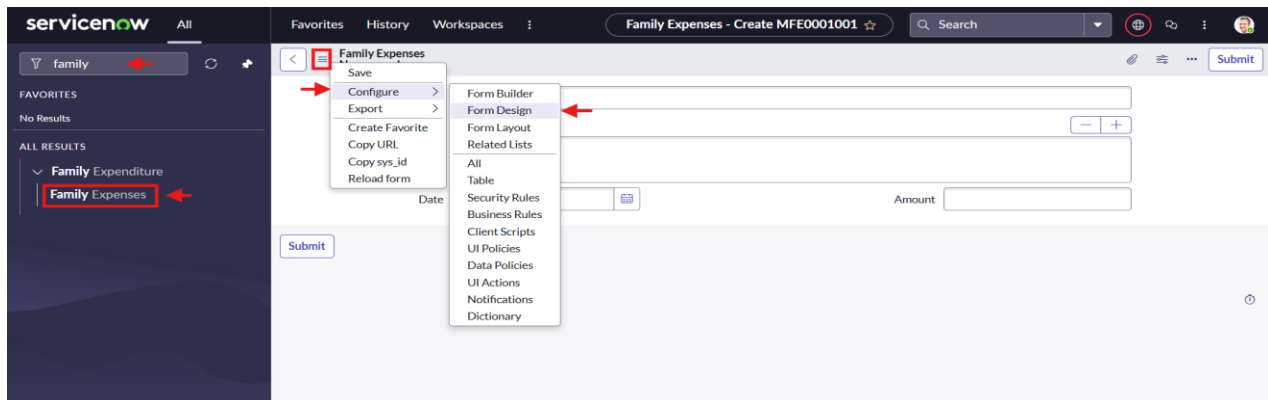
Activity 4: Create Number Maintenance

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**.

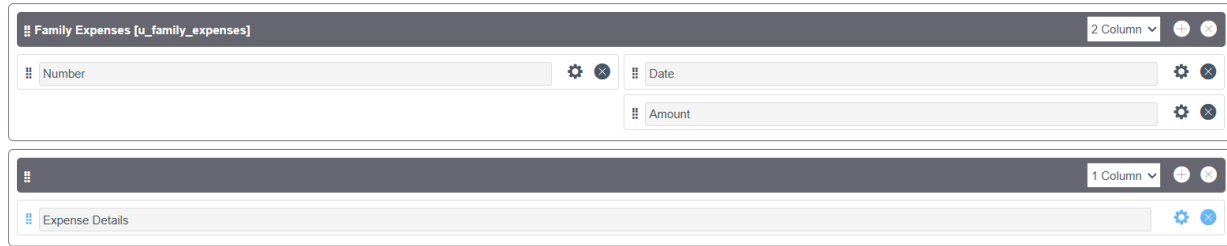


Activity 5: Configure the Form

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**.



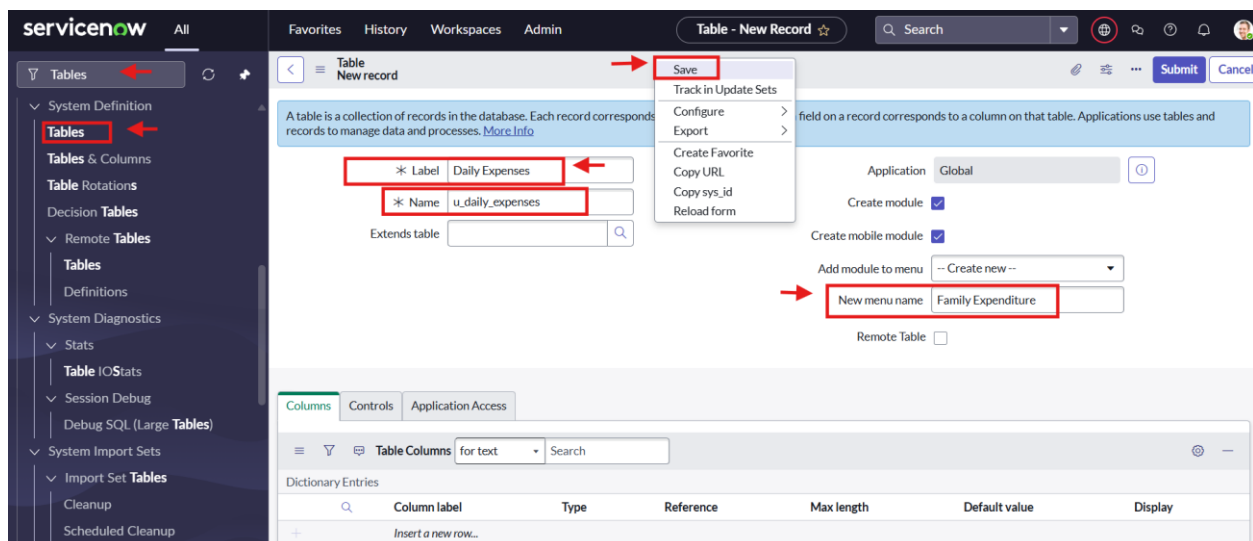
4. Arrange or drag & drop fields as needed.
5. For **Number** field: set it to **Read-Only** (using the gear icon).
6. For **Date** and **Amount** fields: set them as **Mandatory** (using the gear icon).
7. Click **Save**.



Milestone 4 : Creation of Daily Expenses Table

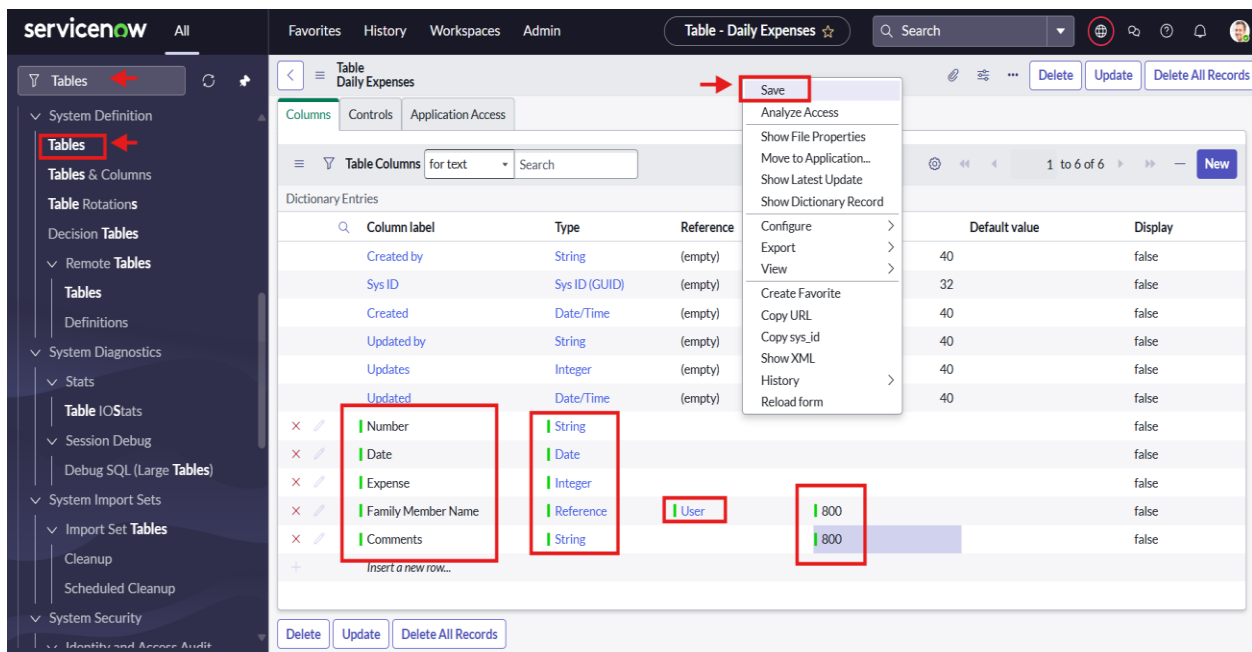
Activity 1: Create Daily Expenses Table

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	Default value	Display
Created by	String	(empty)	40	false
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
Number	String			false
Date	Date			false
Expense	Integer			false
Family Member Name	Reference	User		false
Comments	String		800	false

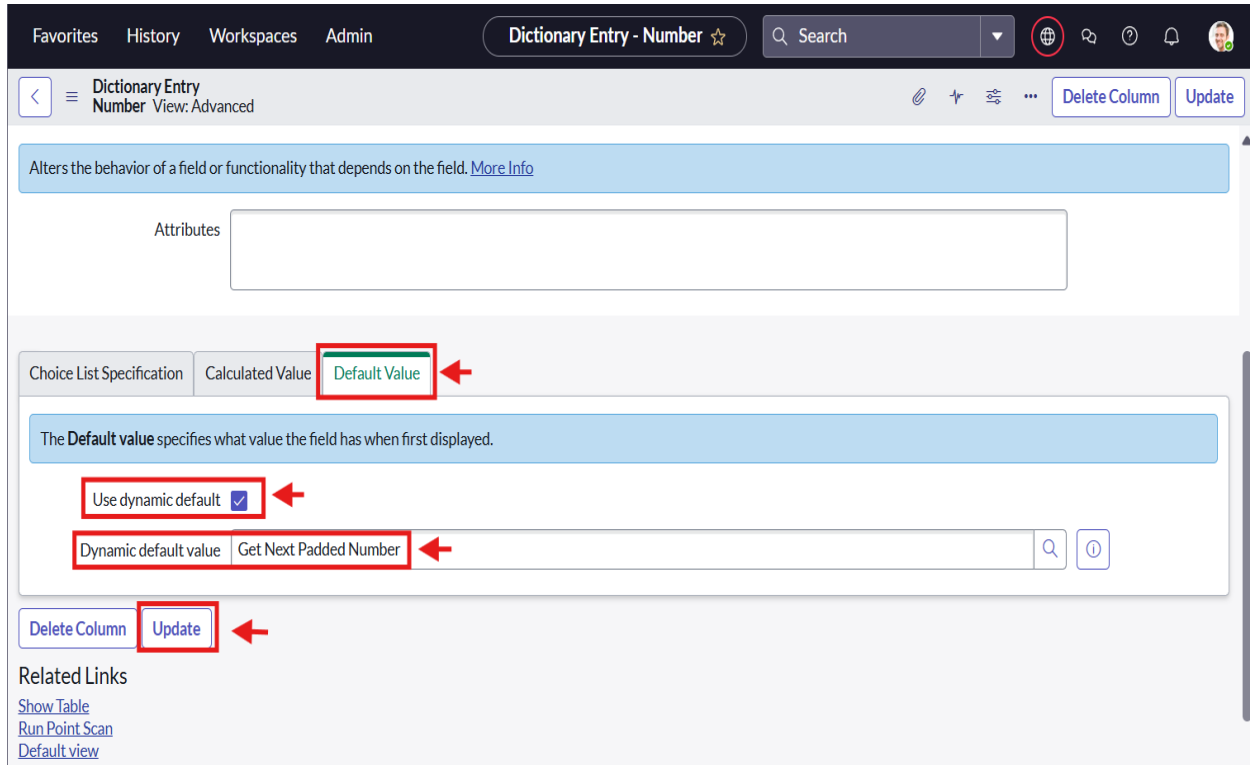
Activity 3: Make Number Field Auto-Generate

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

Attributes

Choice List Specification Calculated Value **Default Value**

The **Default value** specifies what value the field has when first displayed.

Use dynamic default ☒

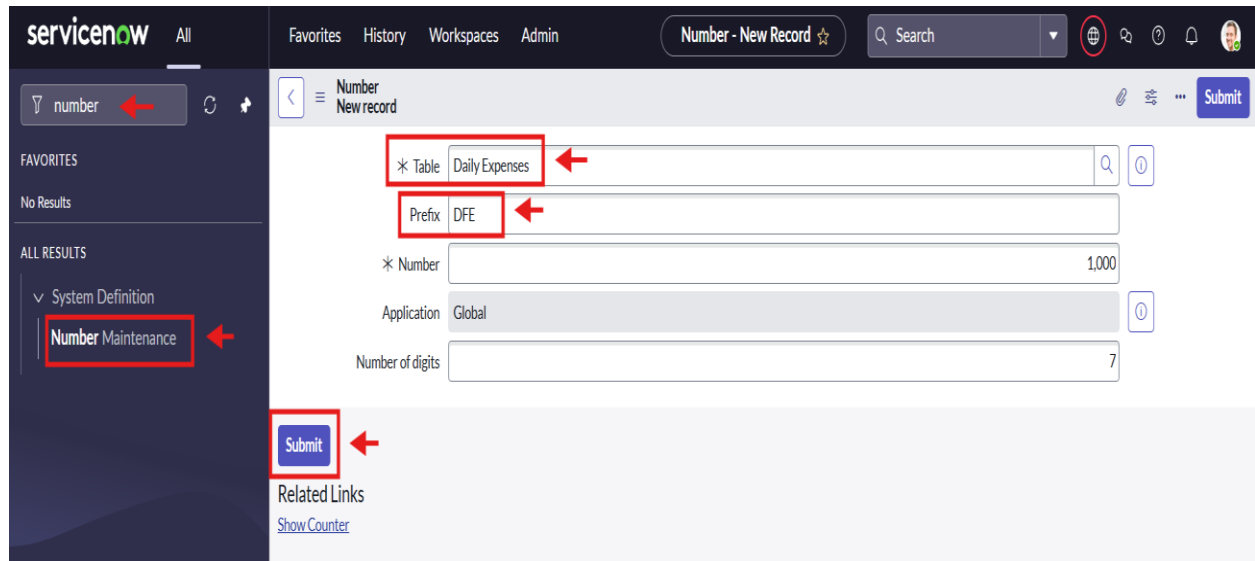
Dynamic default value Get Next Padded Number

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 All Favorites History Workspaces Admin Number - New Record

number

FAVORITES No Results

ALL RESULTS System Definition Number Maintenance

* Table Daily Expenses

Prefix DFE

* Number 1,000

Application Global

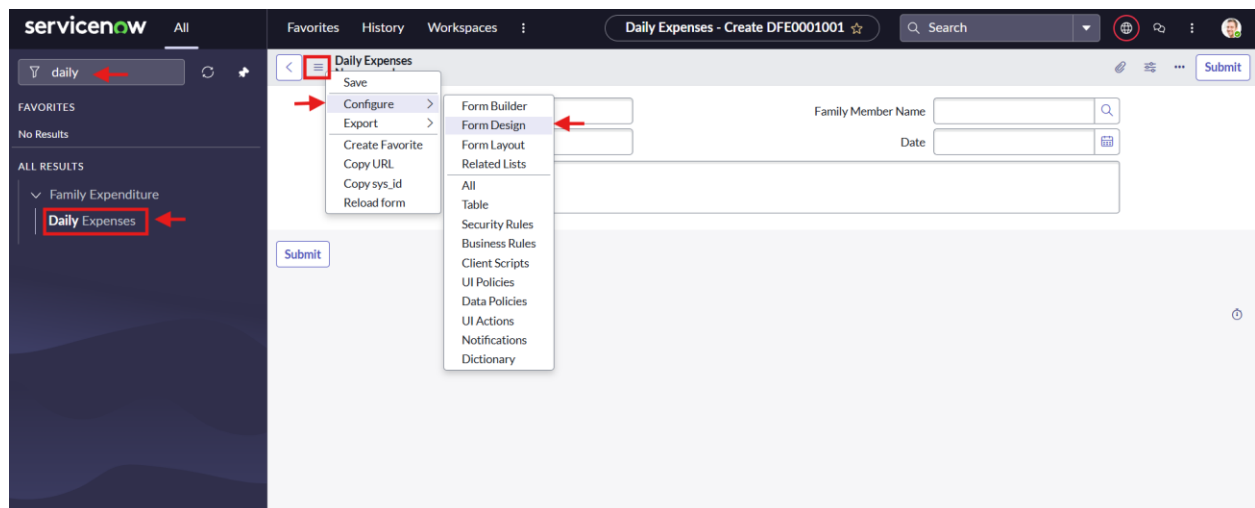
Number of digits 7

Submit

Related Links Show Counter

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 All Favorites History Workspaces Daily Expenses - Create DFE0001001

daily

FAVORITES No Results

ALL RESULTS Family Expenditure Daily Expenses

Save

Configure

Export

Create Favorite

Copy URL

Copy sys_id

Reload form

Form Builder

Form Design

Form Layout

Related Lists

All

Table

Security Rules

Business Rules

Client Scripts

UI Policies

Data Policies

UI Actions

Notifications

Dictionary

Family Member Name

Date

Submit

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**.

Daily Expenses [u_daily_expenses]
2 Column

Number	Family Member Name
Date	Expense

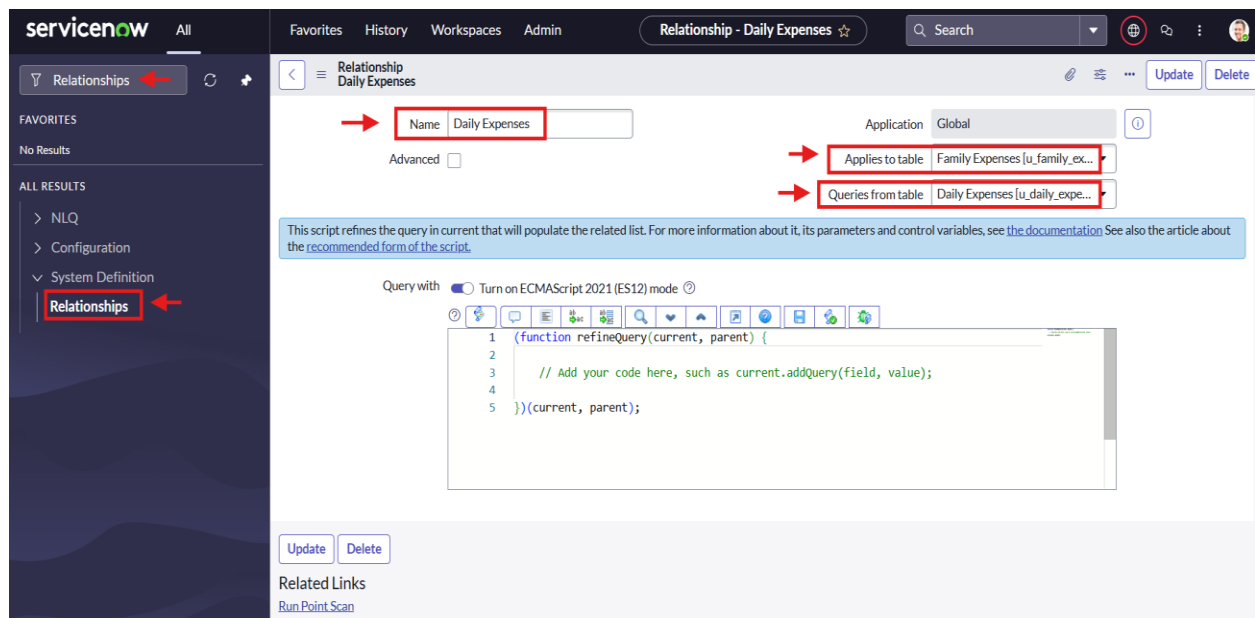
1 Column

Comments

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**.



The screenshot shows the ServiceNow interface for creating a new relationship. The left sidebar has 'Relationships' highlighted. The main form is titled 'Relationship - Daily Expenses'. The 'Name' field is set to 'Daily Expenses'. The 'Applies to table' field is set to 'Family Expenses [u_family_ex...]'. The 'Queries from table' field is set to 'Daily Expenses [u_daily_expe...]'. The 'Application' is set to 'Global'. Below the form, there is a section for 'Query with' and a code editor with the following JavaScript code:

```

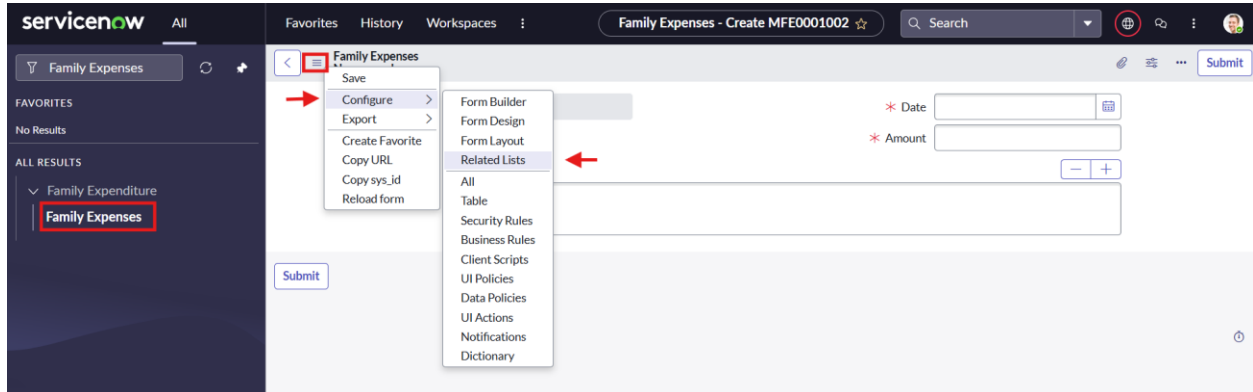
1 (function refineQuery(current, parent) {
2
3   // Add your code here, such as current.addQuery(field, value);
4
5 })(current, parent);
  
```

At the bottom, there are 'Update' and 'Delete' buttons, and a 'Related Links' section with a link to 'Run Point Scan'.

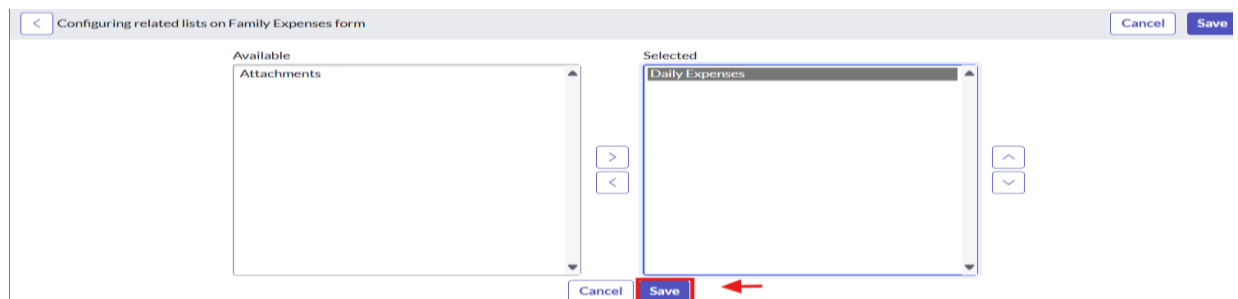
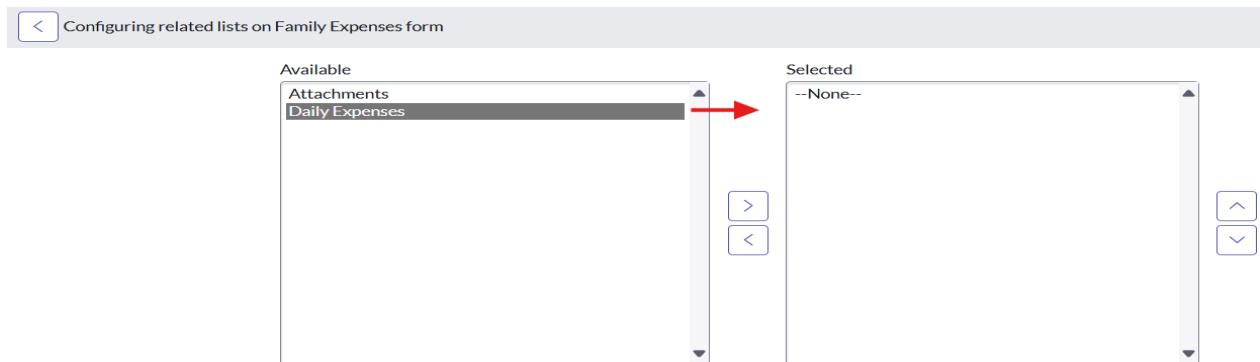
Milestone 6 : Configuring Related List on Family Expenses

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**.



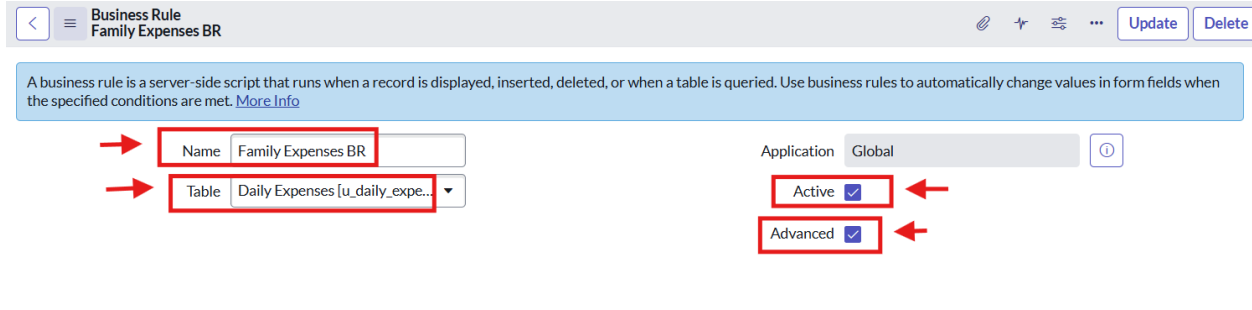
4. From the options, add **Daily Expenses** to the selected area.
5. Click **Save**.



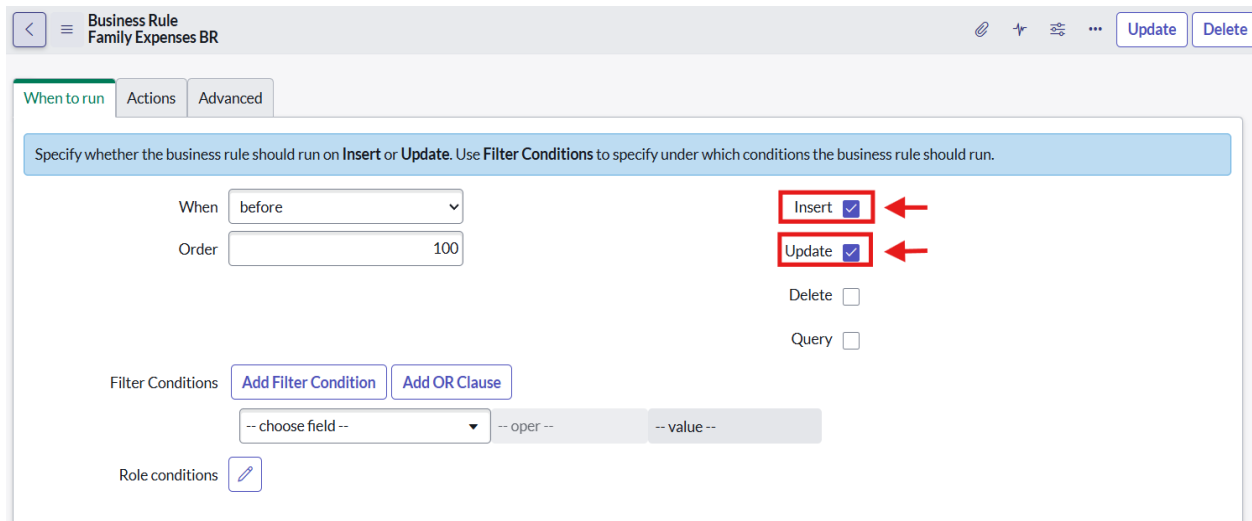
Milestone 7 : Creation of Business Rules

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**.



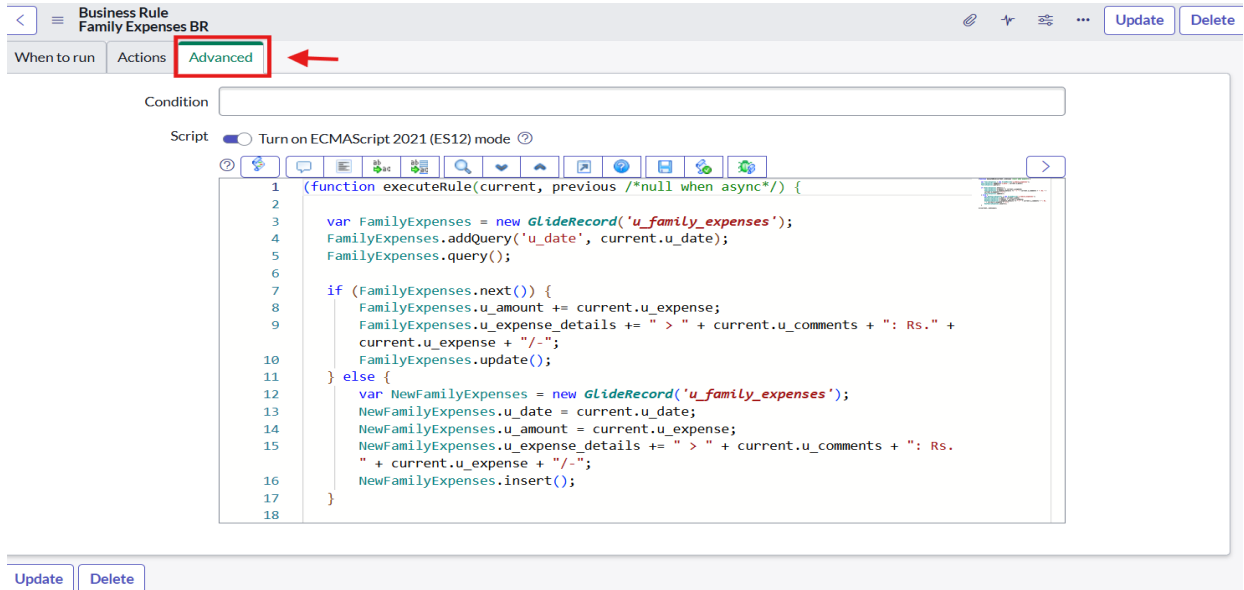
5. Under When to run, check Insert and Update.



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**.

servicenow All

Favorites History Workspaces Admin Relationship - Daily Expenses

Relationships

FAVORITES
No Results

ALL RESULTS
> NLQ
> Configuration
✓ System Definition
 Relationships

Name: Daily Expenses Application: Global
Advanced: ☐ Applies to table: Family Expenses [u_family_ex...
Queries from table: Daily Expenses [u_daily_expe...]

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     // Match Daily Expenses with Family Expenses by Date  
4     current.addQuery('u_date', parent.u_date);  
5     current.query();  
6  
7 })(current, parent);  
8
```

Update **Delete**

Related Links
[Run Point Scan](#)

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.