**CALCULATING FAMILY EXPENSES USING SERVICENOW**

**Team ID: NM2025TMID15854**

**Team Size: 4**

**Team Leader: Vijayalakshmi S**

**Team Member 1: Nandhini Priya P**

**Team Member 2:Nithish Kumar D**

**Team Member 3:Yogeshwaran K**

**PROBLEM STATEMENT**

The project aims to develop a comprehensive expense calculation system using ServiceNow. This system will enable users to track and manage family expenses efficiently. It will include features such as expense categorization, budget setting, real-time tracking, and reporting capabilities. Utilizing ServiceNow's robust platform, the project will ensure seamless integration, user-friendly interface, and scalability to accommodate varying family sizes and financial complexities. The end goal is to empower users with the tools they need to make informed financial decisions and promote financial well-being within the family unit.

**OBJECTIVES**

The objective of this project is to design and implement a **custom ServiceNow application** to track and manage **Family Expenses** efficiently. By using custom tables (Family Expenses & Daily Expenses), business rules, and form designs, the project aims to:

* Automate the calculation of total family expenses from daily entries.
* Ensure accuracy and consistency of financial records.
* Minimize manual effort through real-time updates.
* Demonstrate the use of ServiceNow as a **low-code platform** for building practical business solutions.

### SKILLS REQUIRED: IoT Open Hardware Platforms in Family Expenses (ServiceNow)

1. IoT devices (Arduino, Raspberry Pi, sensors) can automatically send data like electricity or grocery costs to the **Daily Expenses** table. A Business Rule then updates **Family Expenses**, making expense tracking real-time and error-free.

### Data Structures in Family Expenses (ServiceNow)

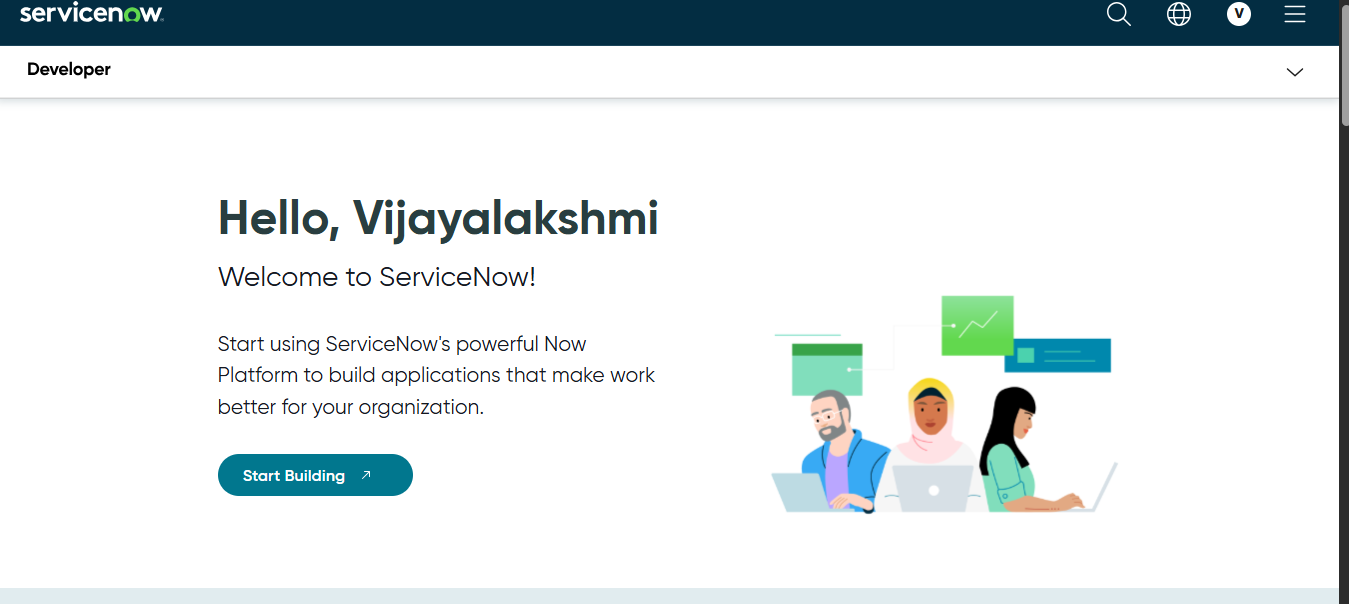
* **Array/List** → Each daily expense record.
* **Hash Map** → Date as key, expenses as value.
* **String** → Expense details appended as text.

**TASK INITIATION**

### Milestone 1: ServiceNow Instance

### Activity 1: Request instance

1. Sign up at [ServiceNow Developer Site](https://developer.servicenow.com?utm_source=chatgpt.com).
2. Log in and go to **Personal Developer Instance.**
3. Click **Request Instance.**
4. Provide required details and submit.
5. Check your email for instance credentials.

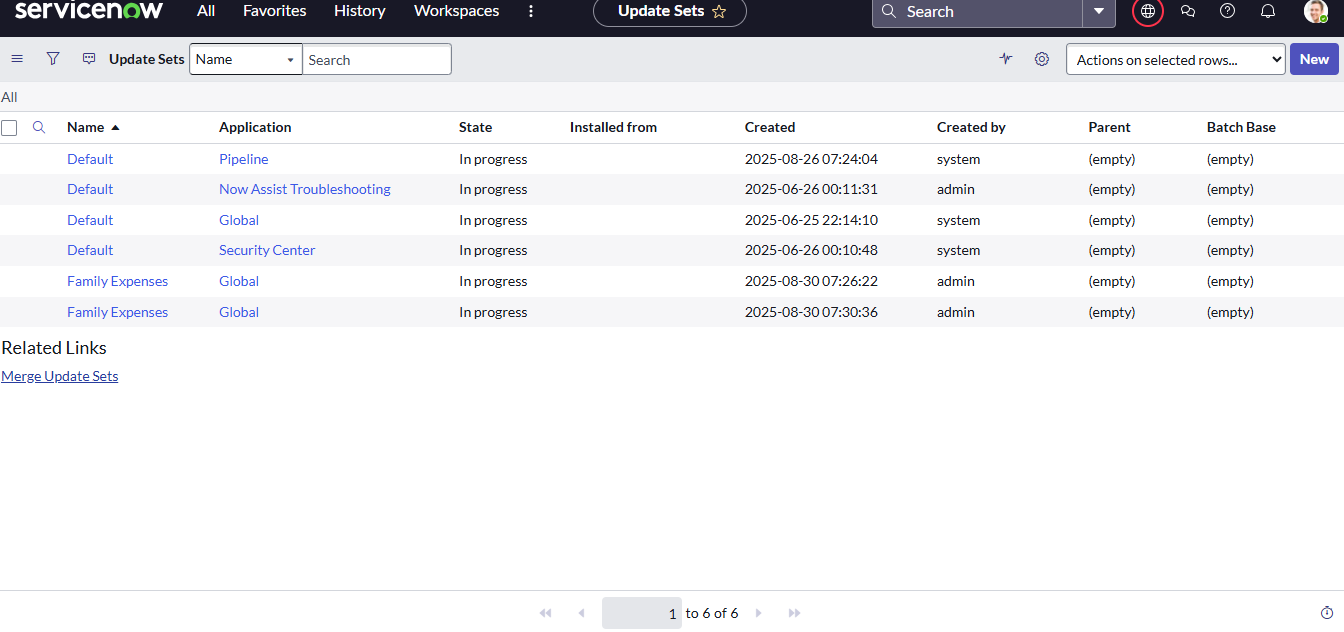


1. Log in to your ServiceNow instance using the credentials.

### Milestone 2: Creation of New Update Set

**Activity 1**:**Update Sets**

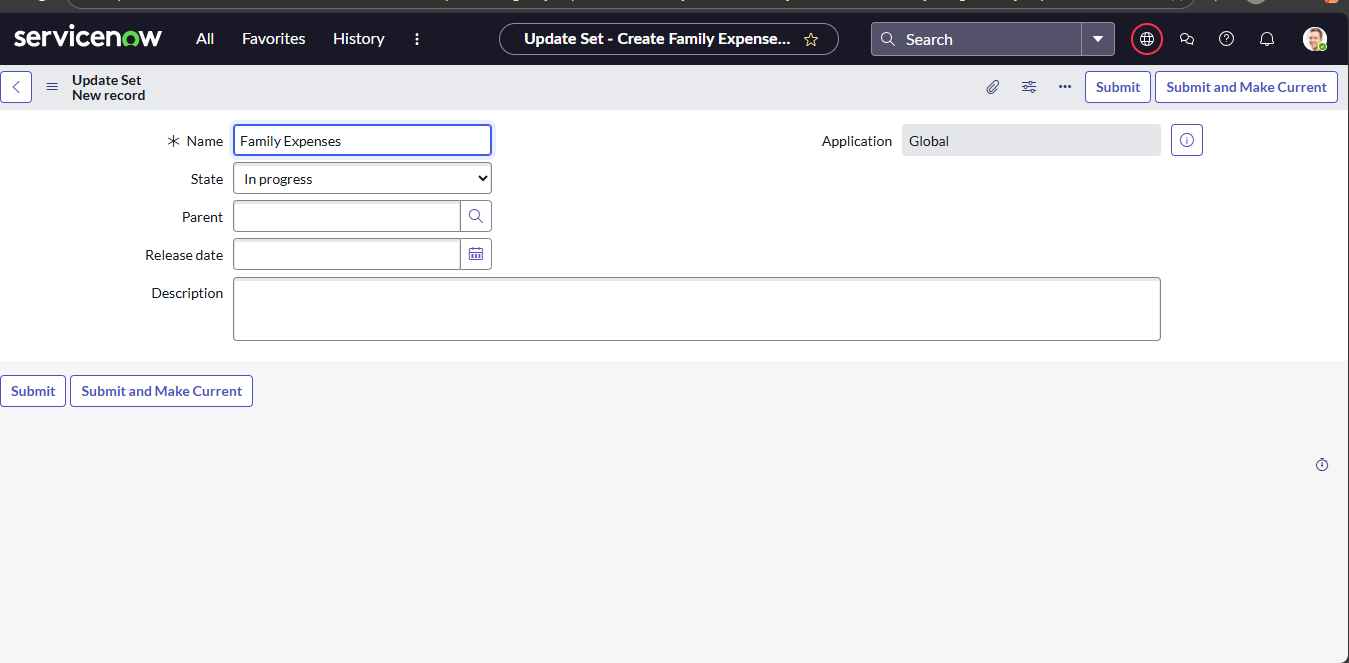
1. Navigate to **All → Local Update Sets** in the filter.
2. Click **New**.



**Figure 2-navigation**

**Enter the details:**

1. Enter **Name: Family Expenses**.
2. Click **Submit.**
3. Select the created Update Set and click **Make Current.**

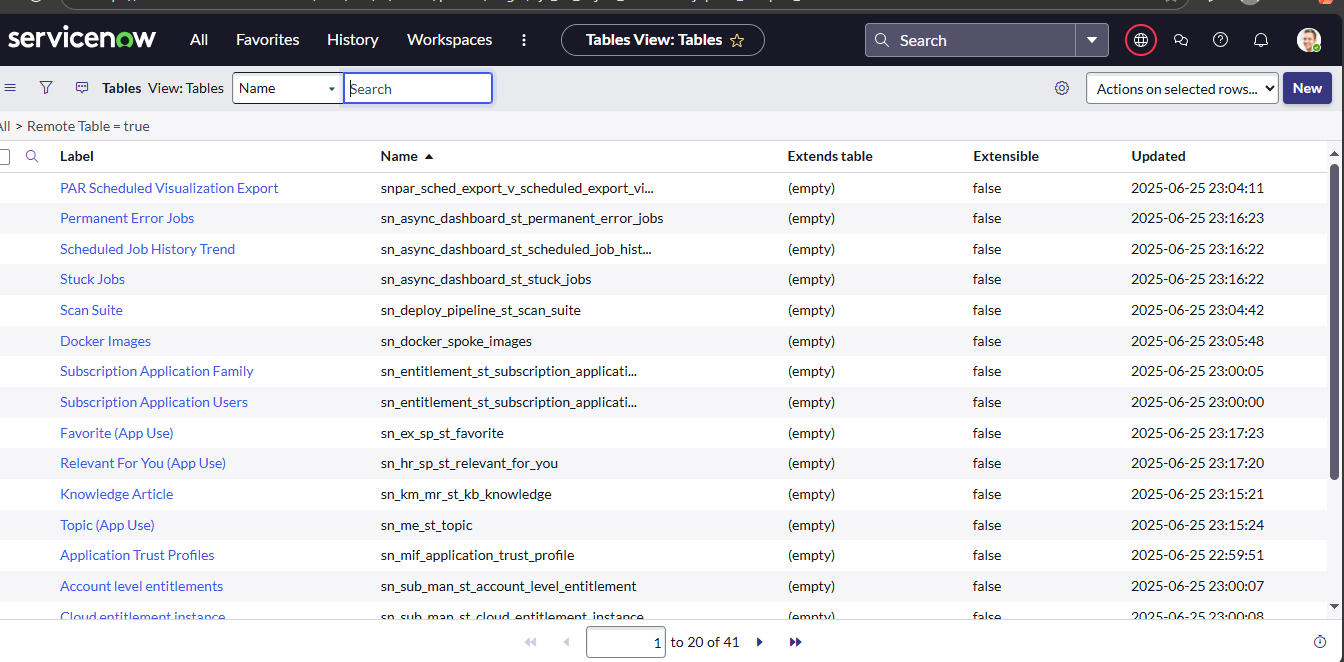


**Figure3- update set**

**Milestone 3: creation of Table**

### Activity 1: Creation of Family Expenses Table

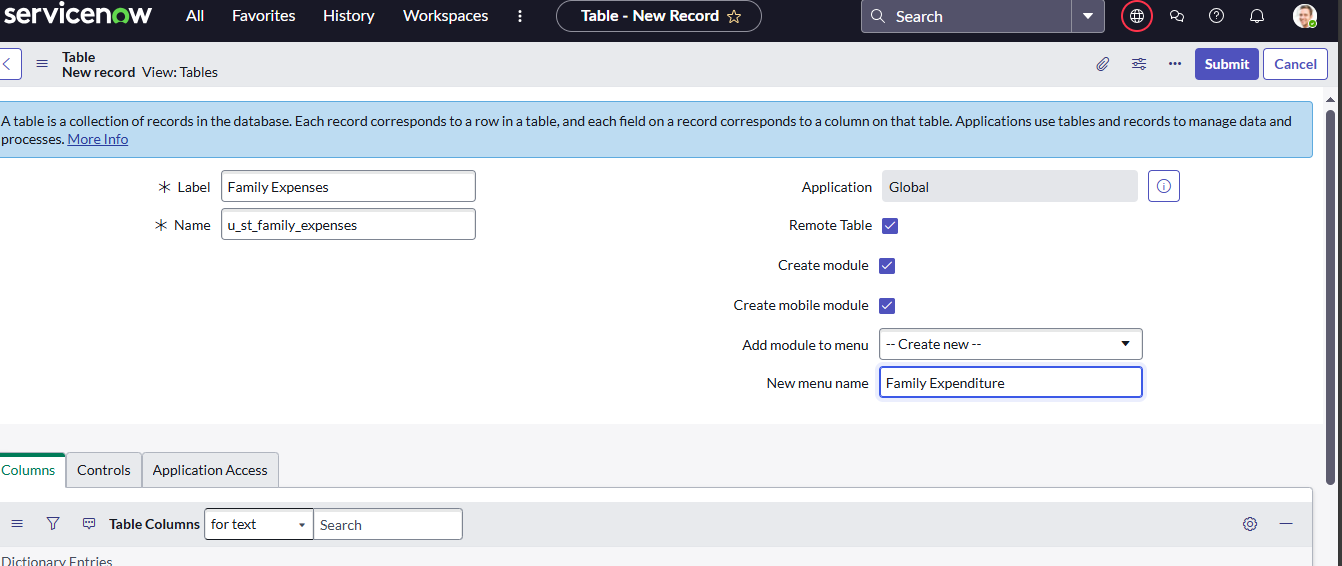
1. Go to **All → Tables** in the filter.
2. Click **New**.



**Figure4-navigate tables**

Enter details:

1. **Label:** Family Expenses
2. **Name:** Auto-Populated
3. **New menu name:** Family Expenditure
4. Right-click on the **Header**.
5. Click **Save**.



### Figure 5- Creation of Family Expenses Table

### Activity 2: Creation of Columns(Fields)

1. In the **Columns** tab, double-click to **insert a new row**.
2. **Column label:** Number
3. **Type:** String

Insert another row:

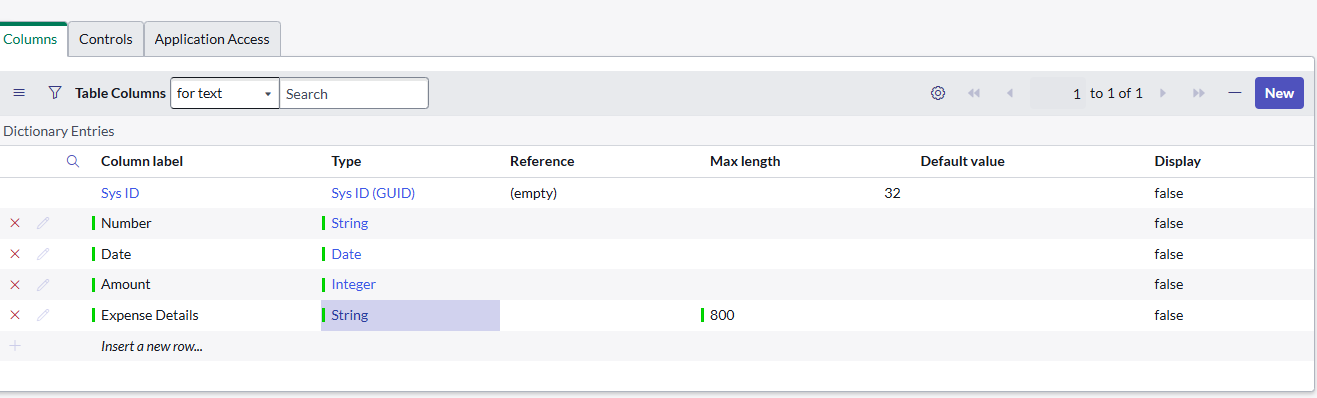
1. **Column label:** Date
2. **Type:** Date

Insert another row:

1. **Column label:** Amount
2. **Type:** Integer

Insert another row:

1. **Column label:** Expense Details
2. **Type:** String
3. **Max length:** 800
4. Go to the **Header**, right-click, and click **Save**

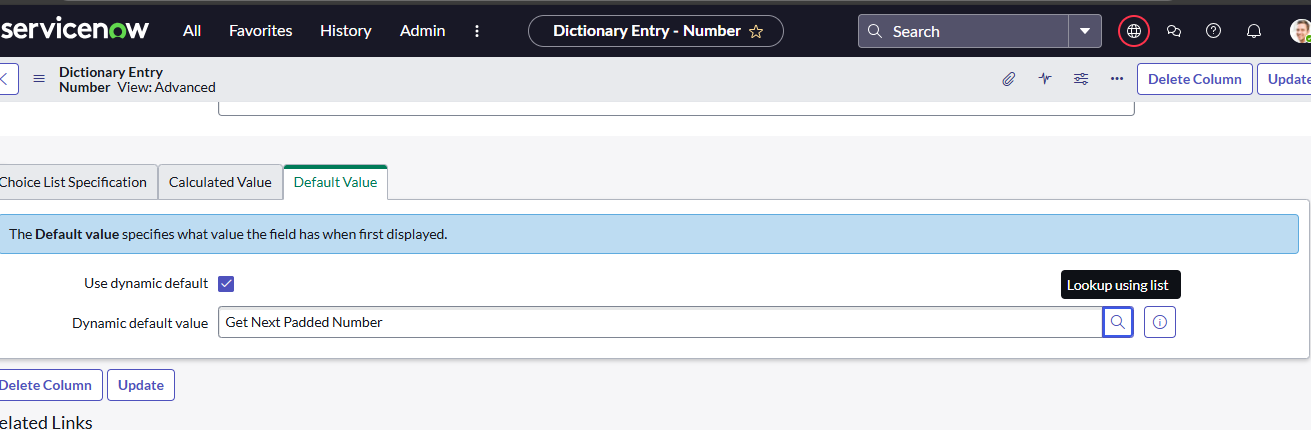


### Figure 6- Creation of Columns(Fields)

### Activity 3: Making Number Field an Auto-Number

**Configure Number Field**

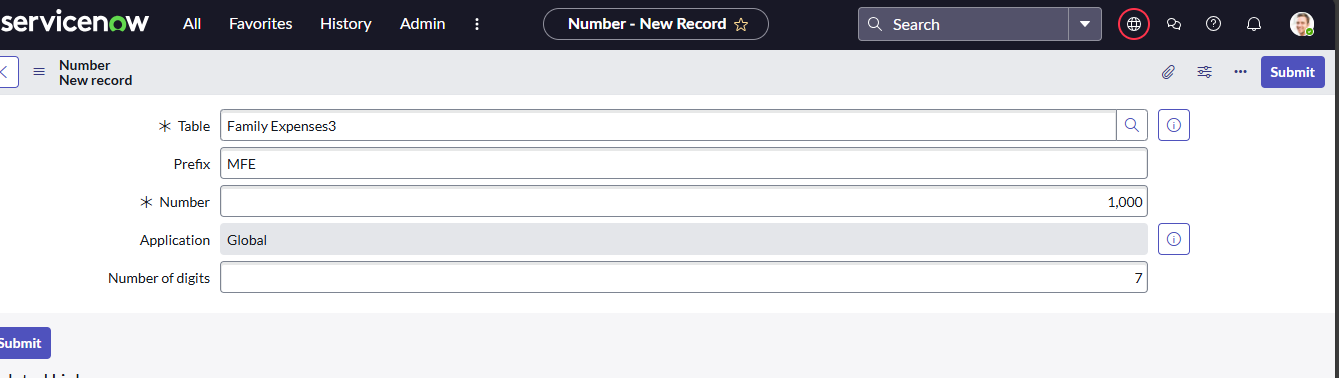
1. Double-click on the **Number** column.
2. Scroll down and open **Advanced View**.
3. Check **Use dynamic default**.
4. Set **Dynamic default value** to **Get Next Padded Number**.
5. Click **Update**.



### Figure 7- Making Number column

**Set Number Maintenance**

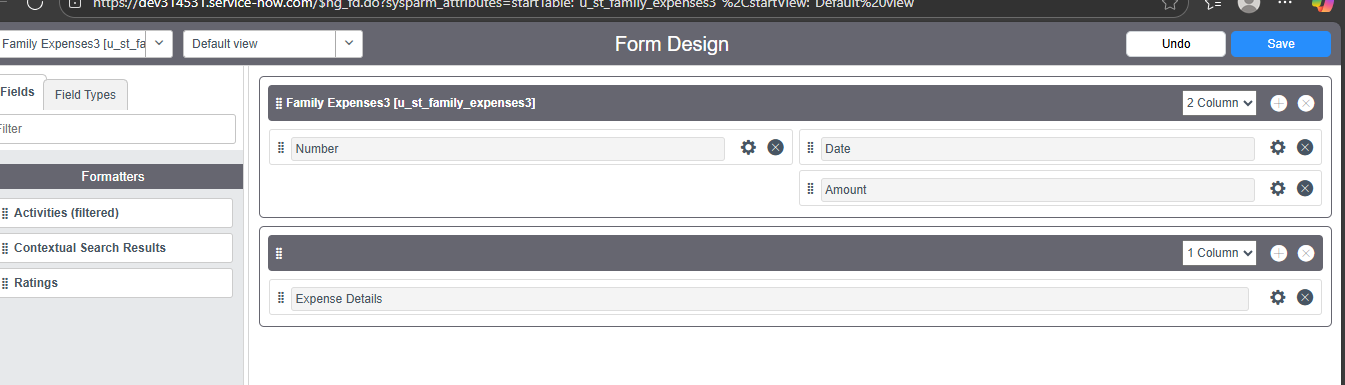
1. Go to **All → Number Maintenance**.
2. Click **New**.
3. Enter:
   * **Table:** Family Expenses
   * **Prefix:** MFE
4. Click **Submit**.



### Figure 8- Making Number Field an Auto-Number

### Activity 4: Configure the Form

1. Go to **All → Family Expenses** and open **Family Expenses**.
2. Click **New**.
3. In the header, right-click → **Configure → Form Design**.
4. Customize the form by drag & drop as needed.
5. For **Number** field → click the gear icon → check **Read-Only**.
6. For **Date** and **Amount** fields → click the gear icon → check **Mandatory**.
7. Click **Save**.

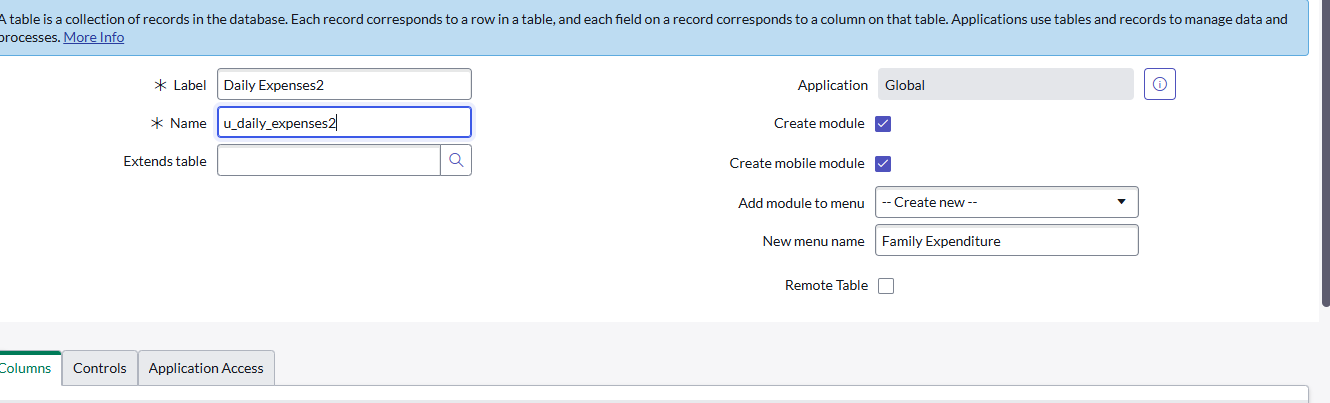


### Figure 9-form design

### Miestone 4: Creation of Table(Daily Expenses)

### Activity 1: Creation of Daily Expenses Table

1. Go to **All → Tables** in the filter.
2. Click **New**.
3. Enter details:
4. **Label:** Daily Expenses
5. **Name:** Auto-Populated
6. **Add Module to menu:** Family Expenditure
7. Go to the **Header**, right-click, and click **Save**



### Figure 10- Creation of Daily Expenses Table

### Activity 2: Creation of Columns(Fields)

In the **Columns** tab, double-click to **insert a new row**.

1. **Column label:** Number
2. **Type:** String

Insert another row:

1. **Column label:** Date
2. **Type:** Date

Insert another row:

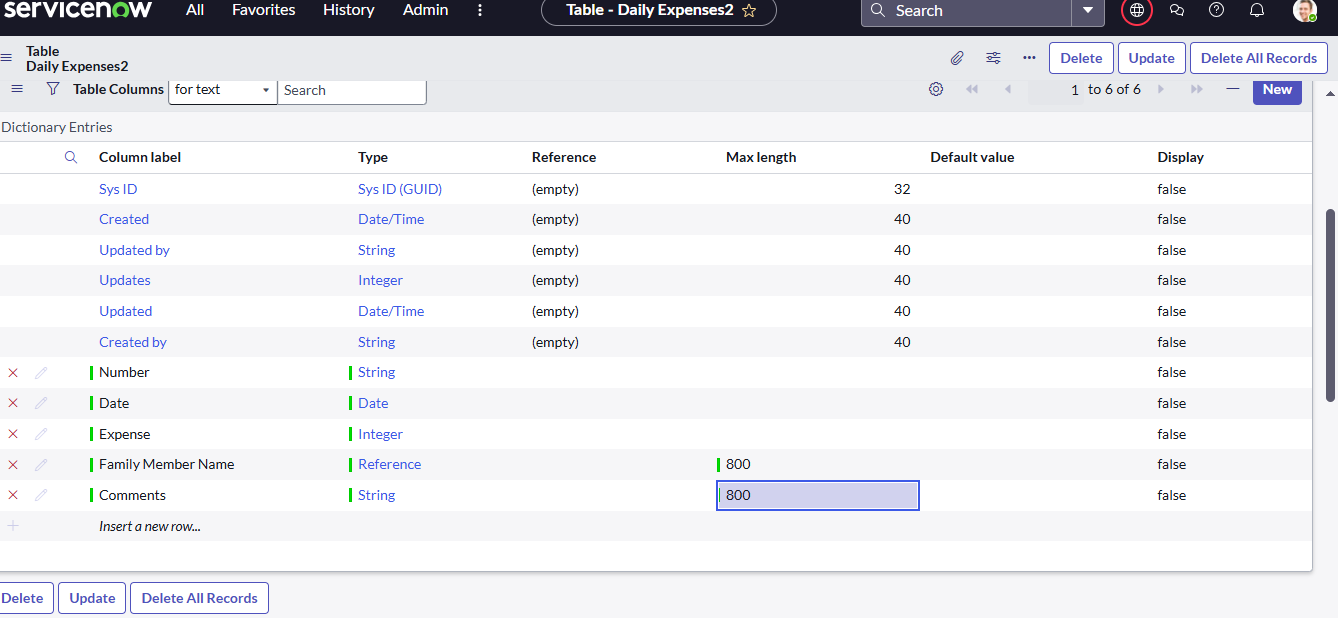
1. **Column label:** Expense
2. **Type:** Integer

Insert another row:

1. **Column label:** Family Member Name
2. **Type:** Reference
3. **Max length:** 800

Insert another row:

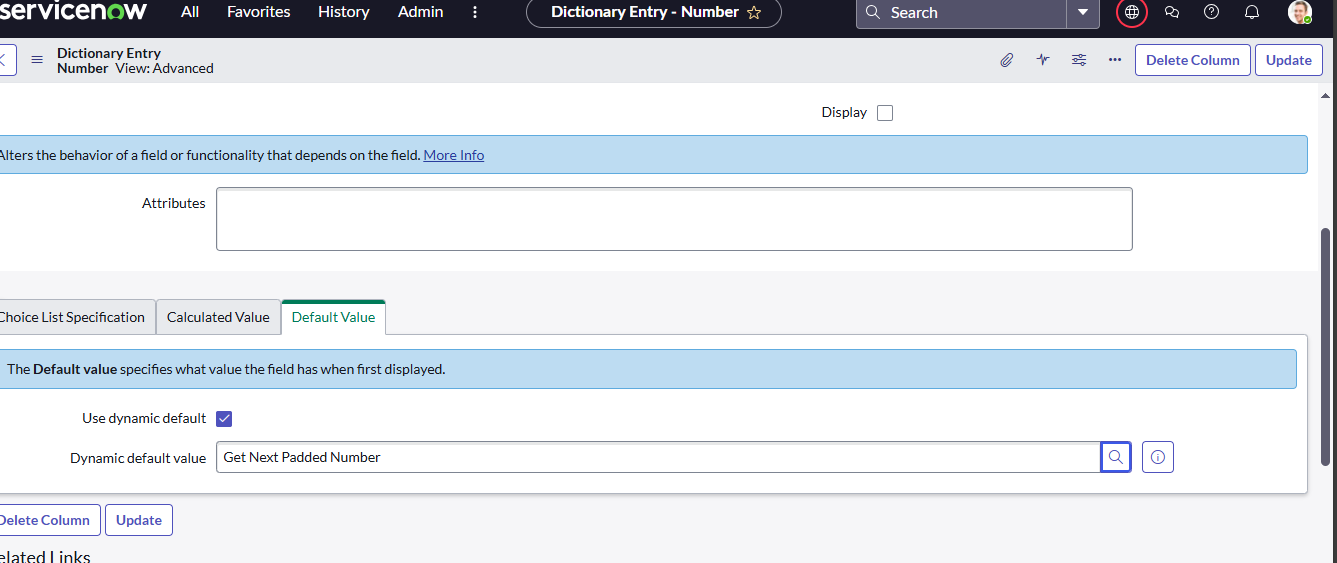
1. **Column label:** Comments
2. **Type:** String
3. **Max length:** 800
4. Go to the **Header**, right-click, and click **Save**.



### Figure 11- Creation of Columns(Fields)

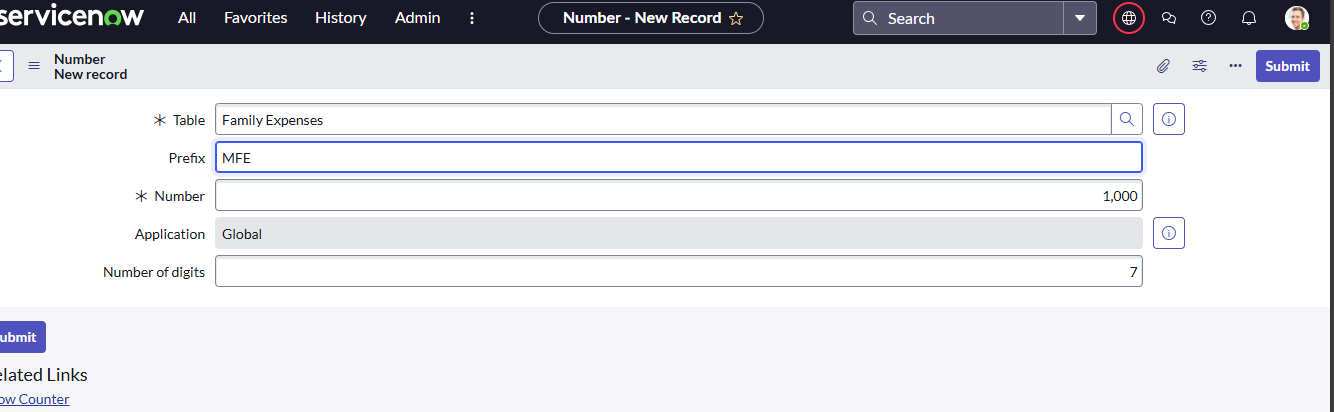
### Activity 3: Making Number Field an Auto-Number

1. **Double-click on the Number column.**
2. **Scroll down and open Advanced View.**
3. **In Default Value:**
4. **Check the box Use dynamic default.**
5. **Set Dynamic default value to Get Next Padded Number.**
6. **Click Update.**

****

### Figure12- Making Number Field an Auto-Number

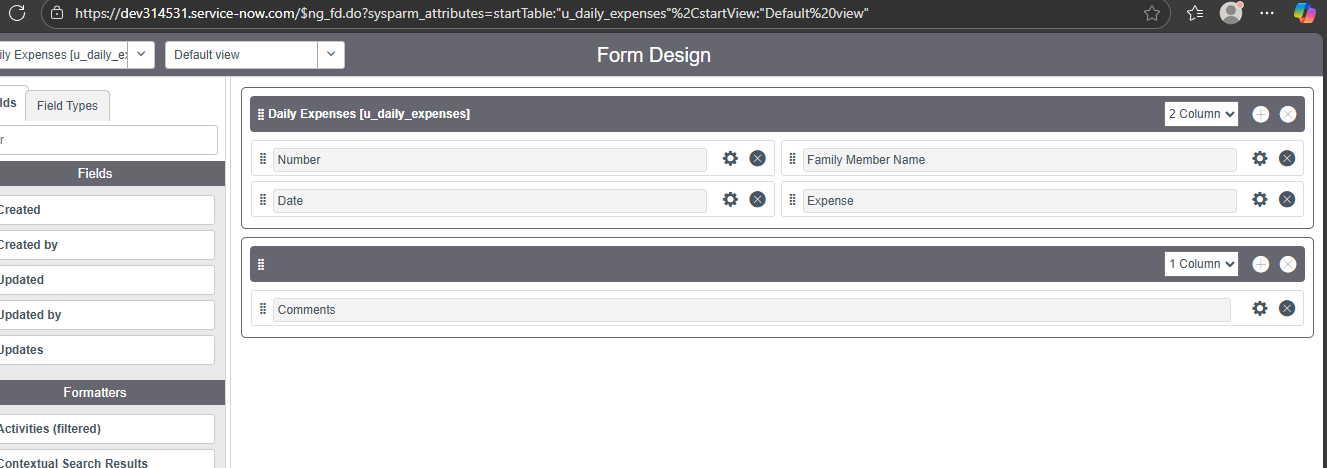
1. Go to **All → Number Maintenance** in the filter.
2. Click **New**.
3. Enter the details:
4. **Table:** Family Expenses
5. **Prefix:** MFE
6. Click **Submit**.



**Figure13-number maintanance**

### Activity 4: Configure the Form

1. Go to **All → Daily Expenses** and open **Daily Expenses**.
2. Click **New**.
3. In the header, right-click → **Configure → Form Design**.
4. Customize the form by drag & drop as needed.
5. For **Number** field → click the gear icon → check **Read-Only**.
6. For **Date** and **Family Member Name** fields → click the gear icon → check **Mandatory**.
7. Click **Save**.

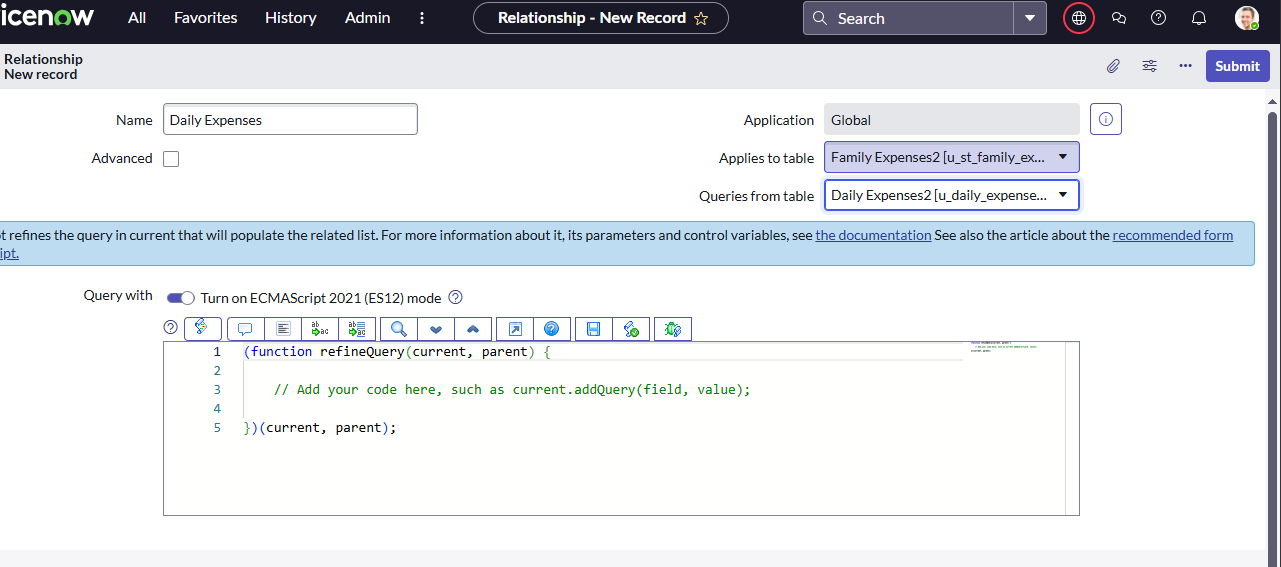
****

### Figure14- Configure the Form

### Milestone 5: Creation of Relationship

### Activity 1: 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:
   1. Name : Daily Expenses
   2. Applies to table : Select Family Expenses
   3. Daily Expenses : Select Daily Expenses
4. Click Save.

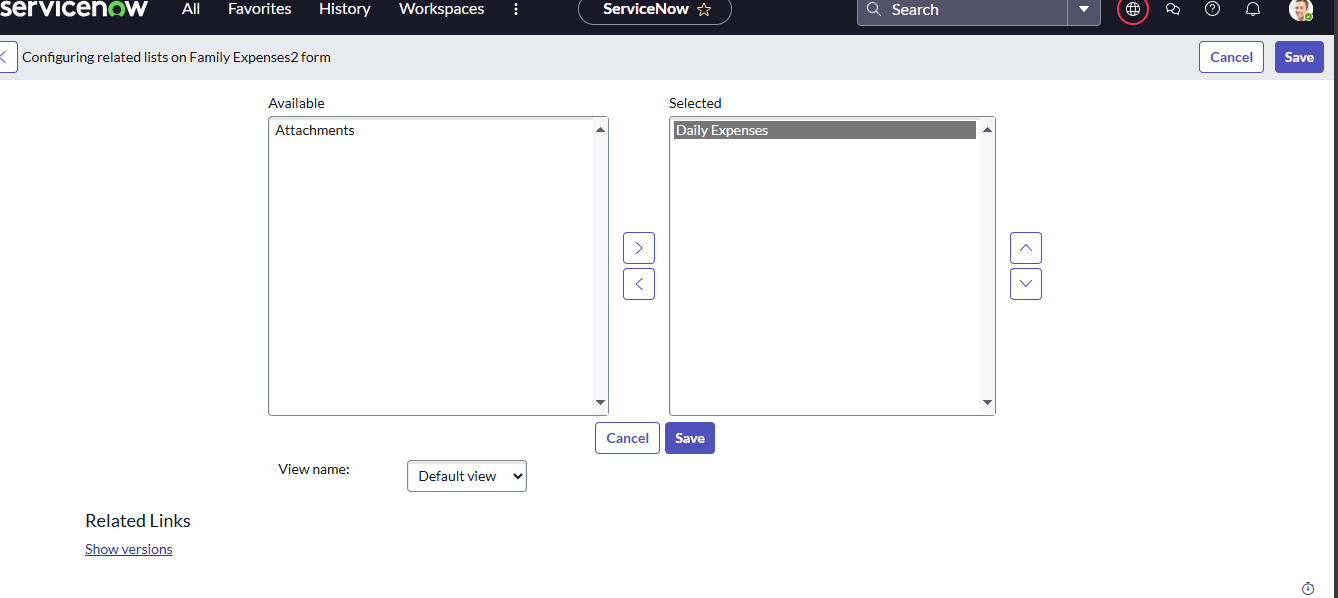


### Figure15- Creation of Relationship

### Milestone 6: Configuring Related List on Family Expenses

### Activity 6: Configuring Related List on Family Expenses

1. Go to **All → Family Expenses** and open **Family Expenses**.
2. Click **New**.
3. In the header, right-click → **Configure → Related Lists**.
4. From the available options, add **Daily Expenses** to the **Selected area**.
5. Click **Save**

****

### Figure16- Configuring Related List on Family Expenses

### Milestone 7: Creation of Business Rules

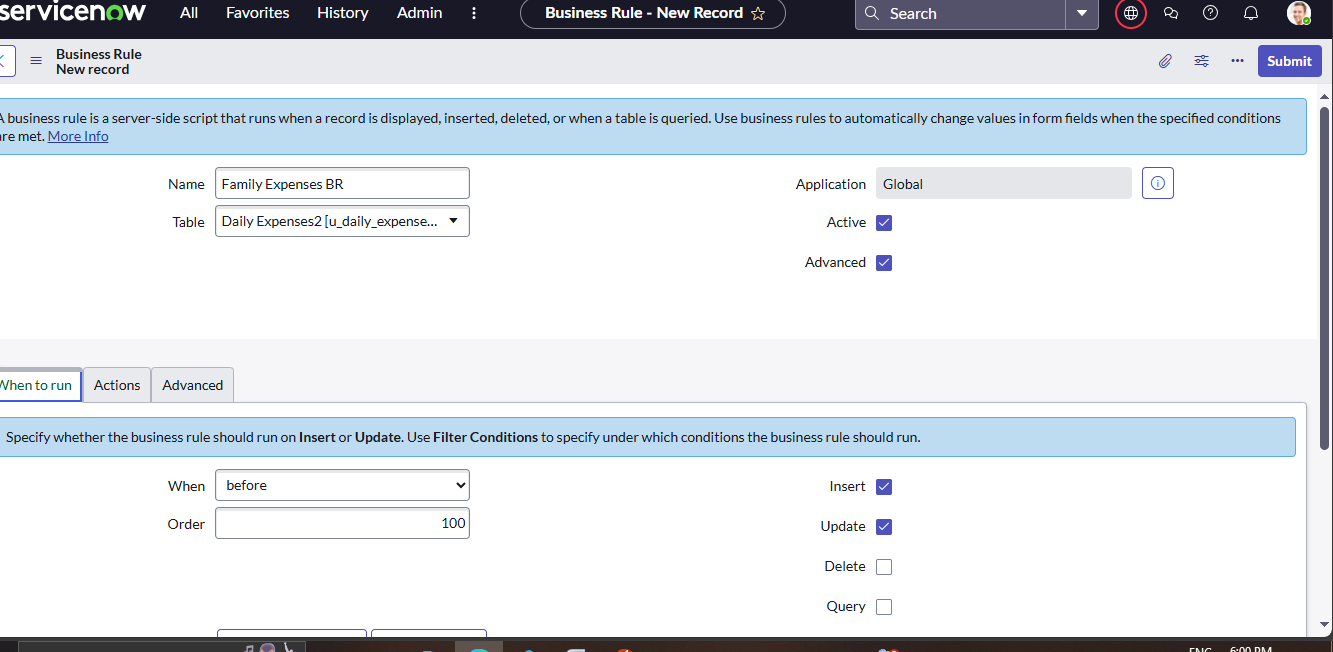
### Activity 7:Business Rules

### Navigate to Business Rules

* Go to **All → Business Rules**.
* Under **System Definition**, select **Business Rules**.
* Click **New**.

### Enter Details

* **Name:** Family Expenses BR
* **Table:** Daily Expenses
* **Check:** **Advanced**
* **When to run:** Select **Insert** and **Update**.



### Figure17- setup

### Add Script (in Advanced section)

Paste the following 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 + "/-";

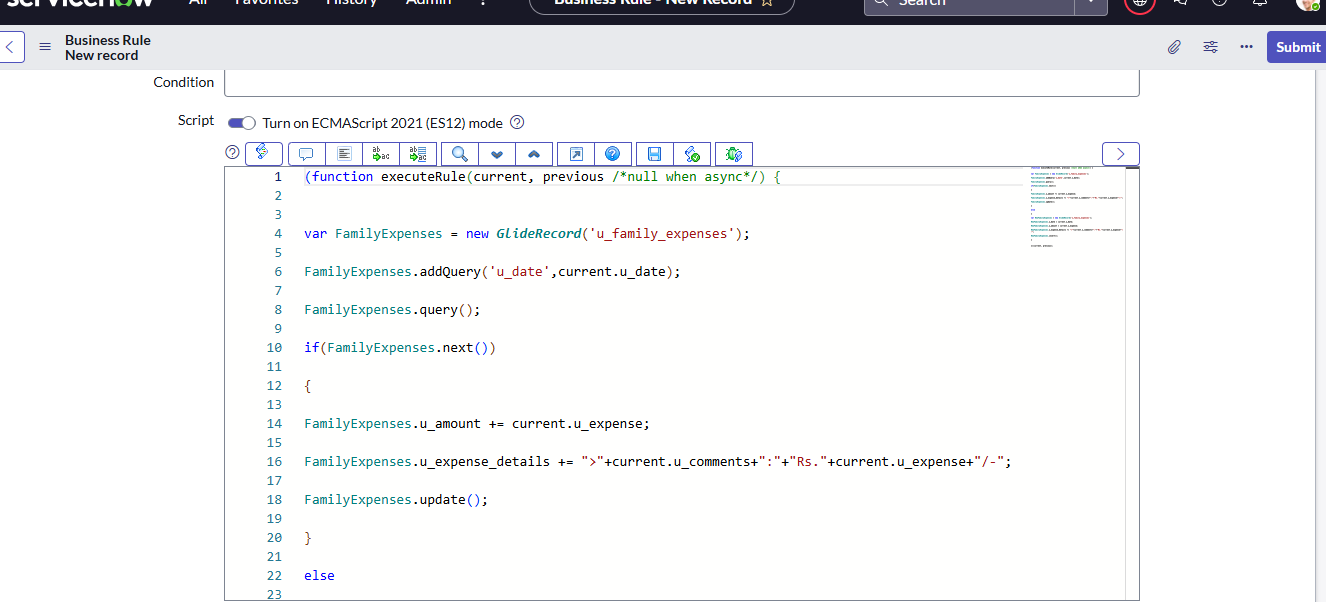
NewFamilyExpenses.insert();

}

})(current, previous);

### Save

* Right-click on the header.
* Click **Save**



### Figure18- Creation of Business Rules

### Milestone 8: Configure the Relationship

### Activity8: 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) {

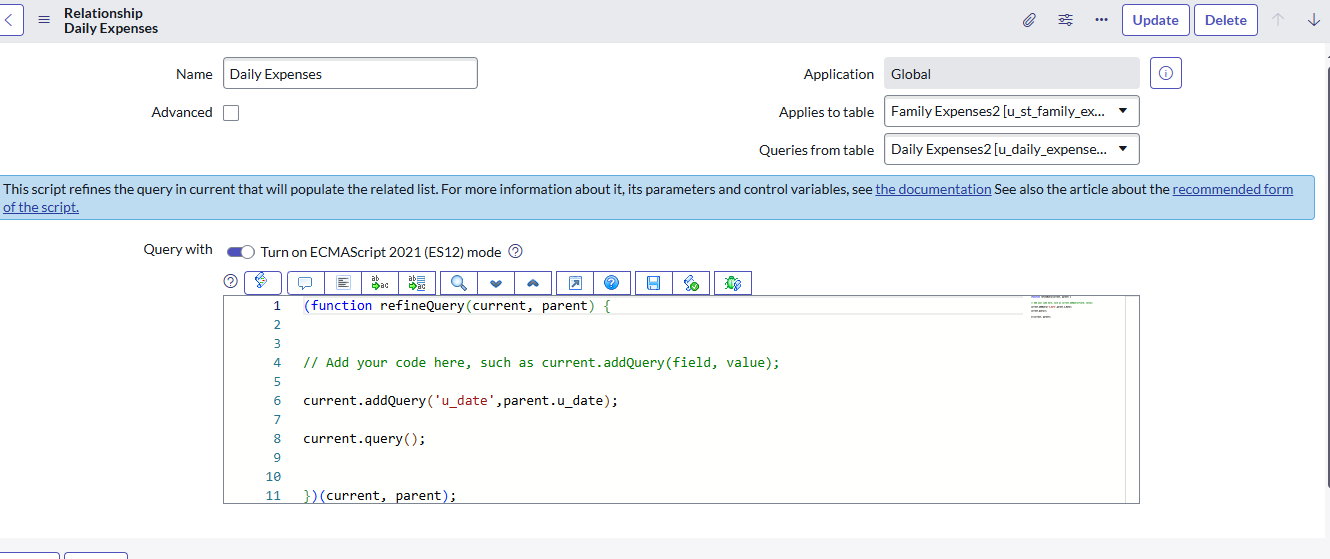
// Add your code here, such as current.addQuery(field, value);

current.addQuery('u\_date',parent.u\_date);

current.query();

})(current, parent);

1. Click on Update.

****

### Figure19- Configure the Relationship

**CONCLUSION**

Implementing the **Family Expenses** and **Daily Expenses** applications in ServiceNow demonstrates how the platform can be customized to handle **real-time calculations and automation**. By creating custom tables, forms, related lists, and business rules, we enabled automatic updates of total family expenses whenever new daily expenses are added. This not only reduces manual effort but also ensures accuracy, consistency, and transparency of financial records.

Overall, ServiceNow’s **low-code/no-code capabilities** combined with its **automation features** make it a powerful tool for building practical business applications such as expense tracking and reporting.