Title : “Calculating Family Expenses Using Service Now”

Team ID : NM2025TMID13043

Team Size : 4

Team Leader : Manoj E

Team member : Bala Mukesh

Team member : Shanmugasundaram .R

Team member : Sonu.S

Acknowledgement:

We sincerely express our gratitude to our faculty and mentors for their guidance and support throughout this project. Their encouragement, feedback, and valuable insights have been crucial in shaping this work. We also extend our appreciation to the ServiceNow developer community and the resources that helped us implement this project successfully.

Abstract:

This project, “Calculating Family Expenses Using ServiceNow”, aims to develop a web-based solution for managing and tracking family expenses. Unlike manual methods, which are prone to errors and inefficiencies, this project leverages ServiceNow’s robust platform to ensure accuracy, scalability, and automation.

The system provides:

Expense Categorization – grouping of expenses into logical categories.

Budget Setting – helping families define and monitor spending limits.

Real-Time Tracking – allowing users to view expense entries as they are recorded.

Reporting – generating summaries that assist in financial decision-making.

The outcome is an application that empowers users to make better financial choices, reduces errors in expense management, and promotes financial well-being within the family.

Project Description:

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.

Table of Contents:

1. Introduction:

Financial planning is an important aspect of every household. Families often struggle to monitor daily expenses, balance income against expenditure, and identify areas of overspending. While spreadsheets and notebooks have traditionally been used for this purpose, they lack automation and scalability.

ServiceNow, a cloud-based platform, provides powerful tools to build applications with database management, automation, and user-friendly interfaces. By leveraging ServiceNow, this project demonstrates how expense tracking can be streamlined into a digital system that is reliable and efficient.

2. Problem Statement:

In many households, expense management is performed manually. This approach introduces challenges such as:

Lack of accuracy in tracking daily transactions

Difficulty in consolidating expenses from multiple family members

No real-time reporting or monitoring

Absence of automation for recurring entries

To address these challenges, the project proposes a ServiceNow-based solution that enables families to record, categorize, and analyze expenses with minimal effort.

3. Objectives:

The main objectives of this project are:

1. To design a family expense management system on the ServiceNow platform.

2. To automate the process of recording expenses through business rules.

3. To establish relationships between daily expenses and overall family expenditure.

4. To ensure the system is scalable for different family sizes.

5. To create a user-friendly interface for recording and viewing expenses.

6. To generate expense summaries and reports for decision-making.

4. System Requirements:

4.1 Hardware Requirements

A standard PC or Laptop with at least 4GB RAM

Internet connection (broadband recommended)

4.2 Software Requirements

ServiceNow Developer Instance (cloud-based)

Web Browser (Chrome, Firefox, or Edge)

GitHub account for version control and code repository

Optional: Word/Excel for exporting data

5. Project Description:

The project leverages ServiceNow's development environment to create a customized application for expense tracking. The system architecture consists of:

Tables: Family Expenses and Daily Expenses for data storage

Fields: To capture details like Date, Amount, Expense Description, and Family Member

Relationships: Linking individual daily expenses with consolidated family expenses

Business Rules: Automating the addition and updating of expenses

Forms: User-friendly data entry interfaces

Reports: Summaries and consolidated views of expenses

This approach ensures that the system is modular, easy to maintain, and extendable for future enhancements.

6. Implementation / Methodology

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

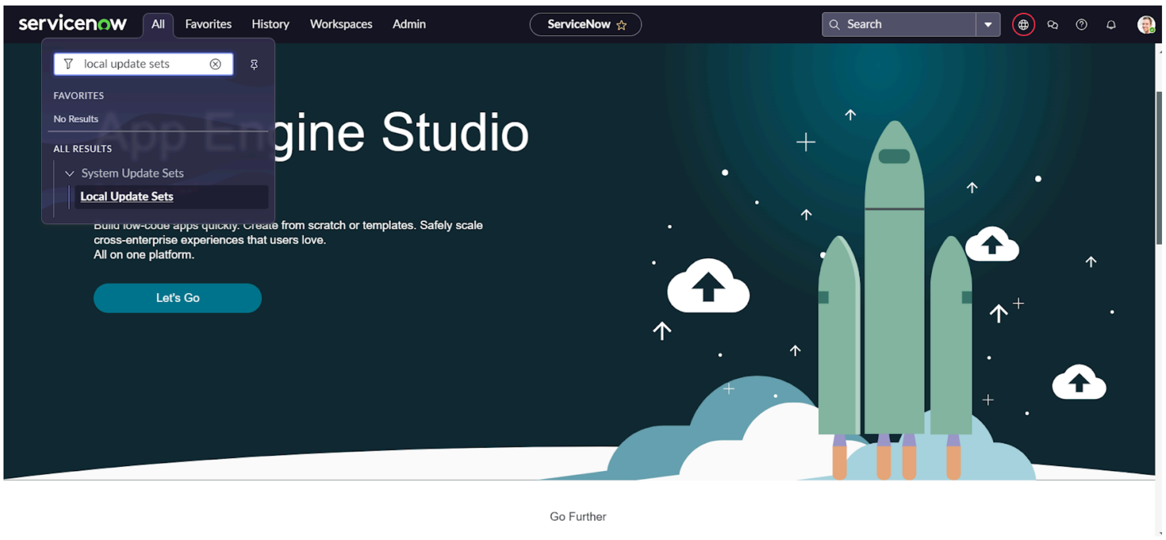
Creation Of New Update Set

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

   Enter the Details as:
   Name : Family Expenses
2. Then click on Submit and Make current.

Creation Of New Update Set

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

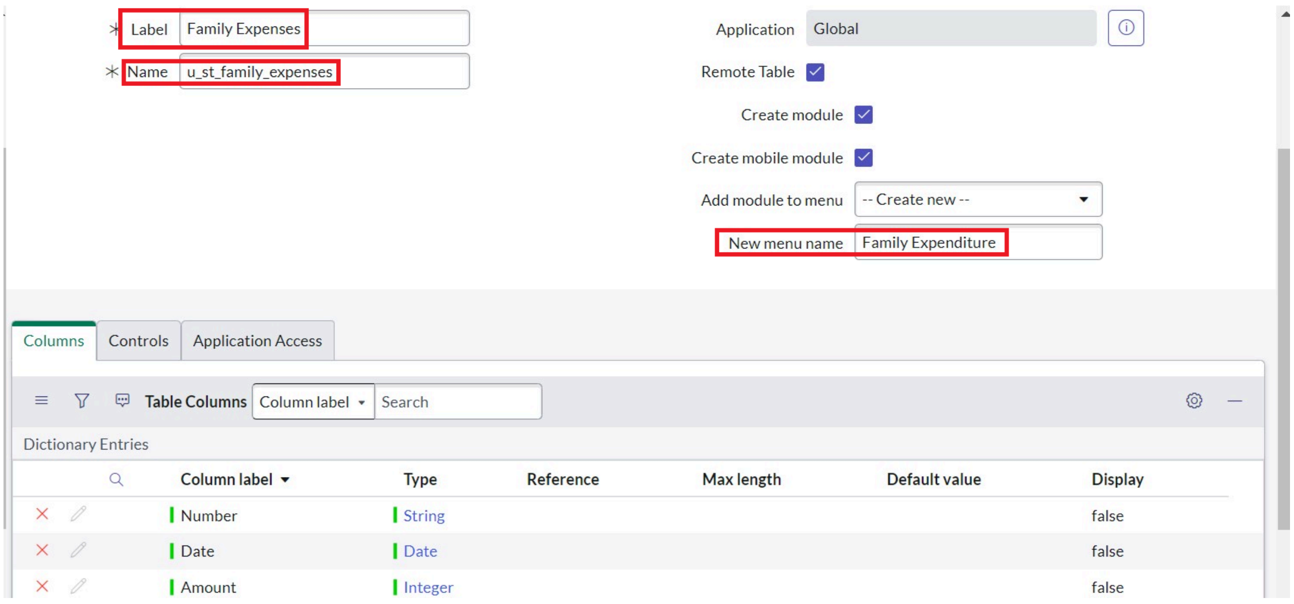


2. Enter the Details as:
   Name : Family Expenses

3. Then click on Submit and Make current.

Creation Of 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



* 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

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

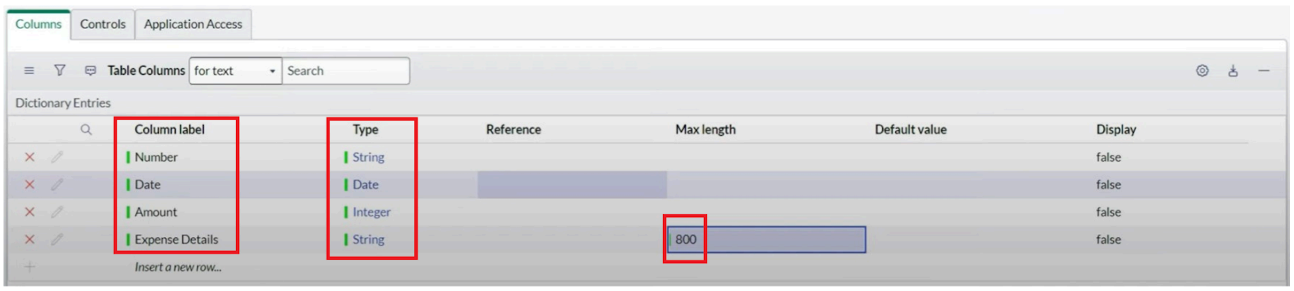

Columns | Controls | Application Access

Table Columns | for text | Search

Dictionary Entries

Column label	Type	Reference	Max length	Default value	Display
Number	String				false
Date	Date				false
Amount	Integer				false
Expense Details	String		800		false
Insert a new row...					

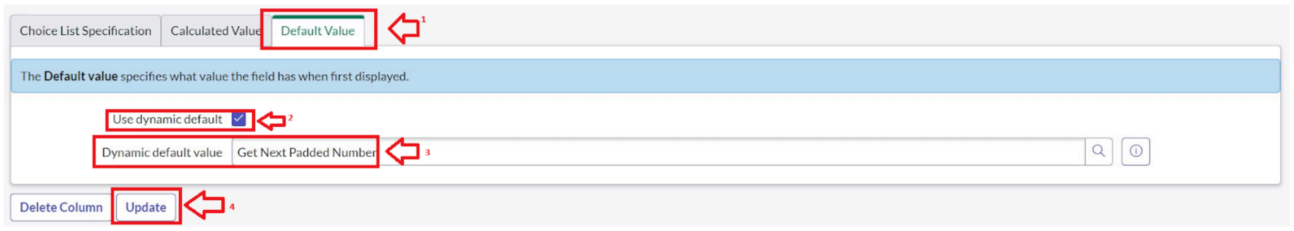
9. Go to the Header. right click there>> click on Save

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

Delete Column | Update

5.

6. Go to All >> In the filter search for Number Maintenance >> select Number Maintenance
7. Click on New.
8. Enter the below Details:
   Table : Family Expenses
   Prefix : MFE

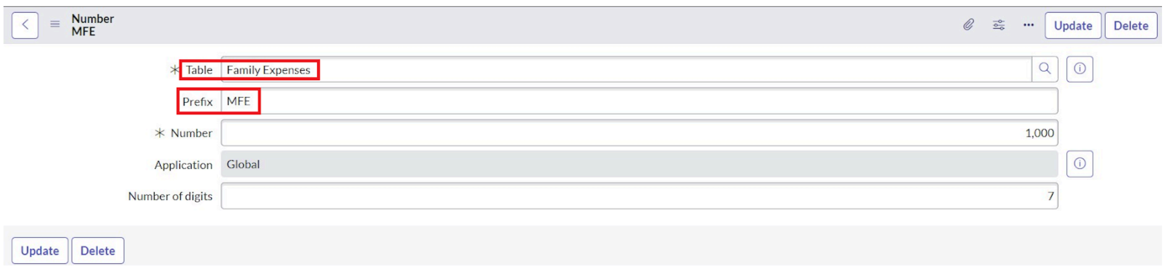
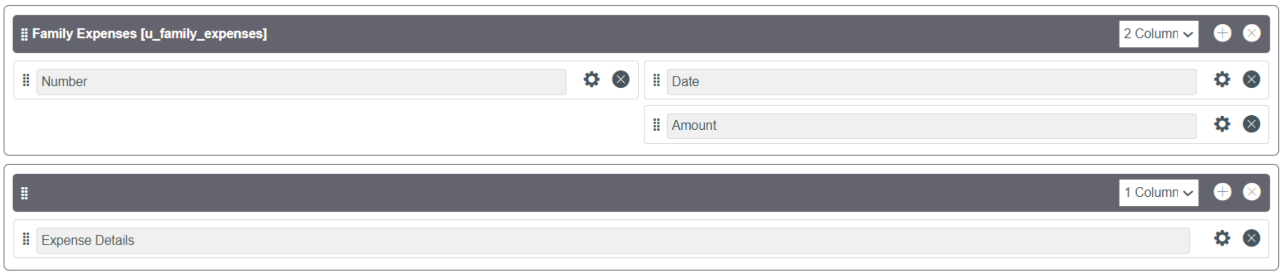


Table	Family Expenses
Prefix	MFE
Number	1,000
Application	Global
Number of digits	7

9. 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.

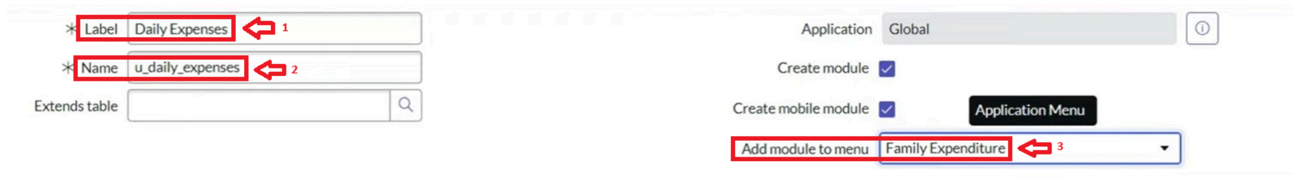


5. Make Number Read-Only Field by clicking on the gear icon and checking Read-Only
6. Make Date, Amount Mandatory Field by clicking on the gear icon and checking Mandatory
7. Click on Save.

Creation Of Daily Expenses Table

1. Go to All > In the filter search for Tables > click on New.
2. Enter the Details:
   Label : Daily Expenses
   Name : Auto-Populated

Add Module to menu : Family Expenditure



Label: Daily Expenses 1
Name: u_daily_expenses 2
Extends table:
Application: Global
Create module: ✓
Create mobile module: ✓
Add module to menu: Family Expenditure 3

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 : Expense
   Type : Integer
7. Double click on insert a new row again
8. Give the details as:

Column label : Family Member Name

Type : Reference
Max length : 800

9. Double click on insert a new row again
10. Give the details as:

Column label : Comments
Type : String
Max length : 800

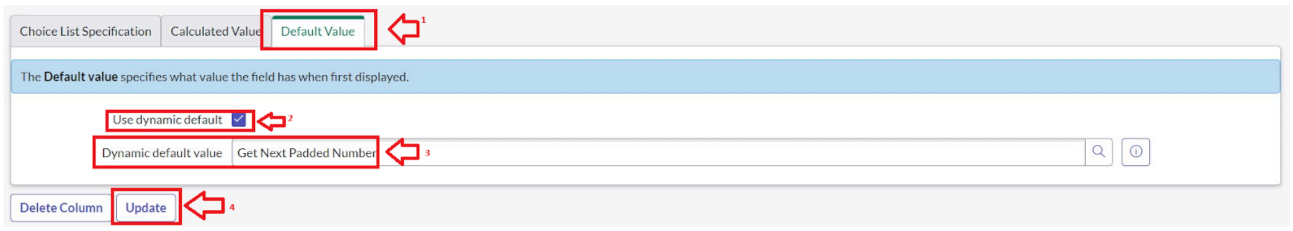
11. Go to the Header and right click there>> click on Save.

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.


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

Use dynamic default ✓

Dynamic default value: Get Next Padded Number

5.
6. Go to All >> In the filter search for Number Maintenance >> select Number Maintenance
7. Click on New.
8. Enter the below Details:

Table : Family Expenses
Prefix : MFE

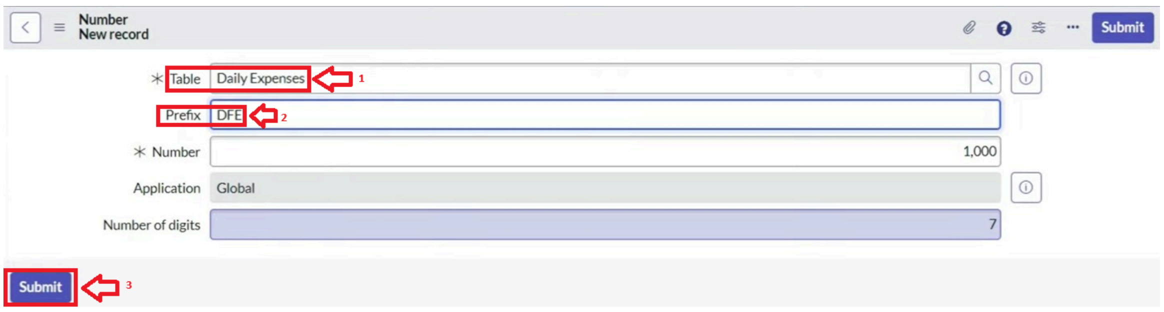


Table: Daily Expenses 1

Prefix: DFE 2

Number: 1,000

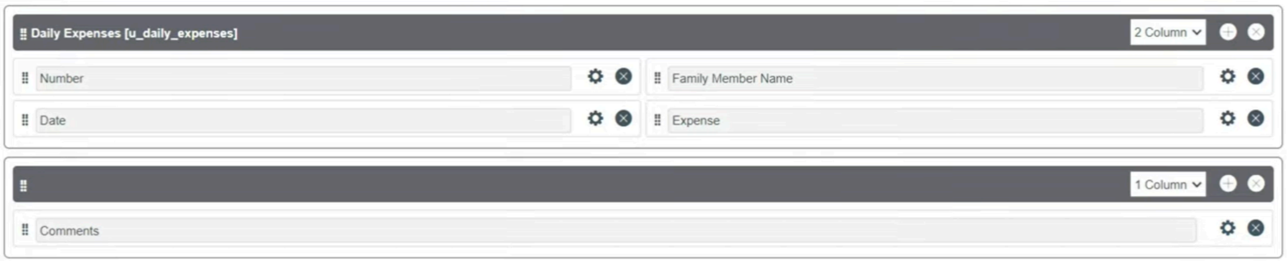
Application: Global

Number of digits: 7

9. Click on Submit.

Configure The Form

1. Go to All >> In the filter search for Daily Expenses >> Open Daily 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.



Daily Expenses [u_daily_expenses]

Number	Family Member Name
Date	Expense
Comments	

5. Make Number Read-Only Field by clicking on the gear icon and checking Read-Only
6. Make Date, Family Member Name 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

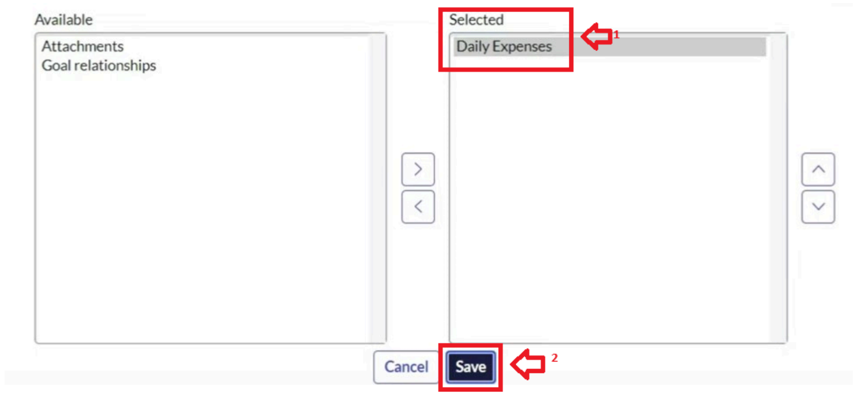
1. Go to All >> In the filter search for Relationships >> Open Relationships
2. Click on New.
3. Enter the details:

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

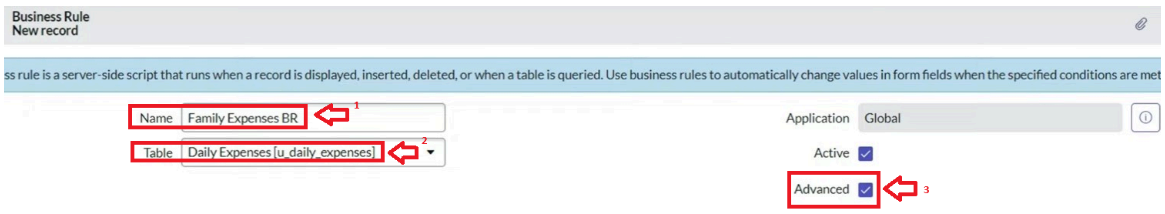


Creation Of Business Rules

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

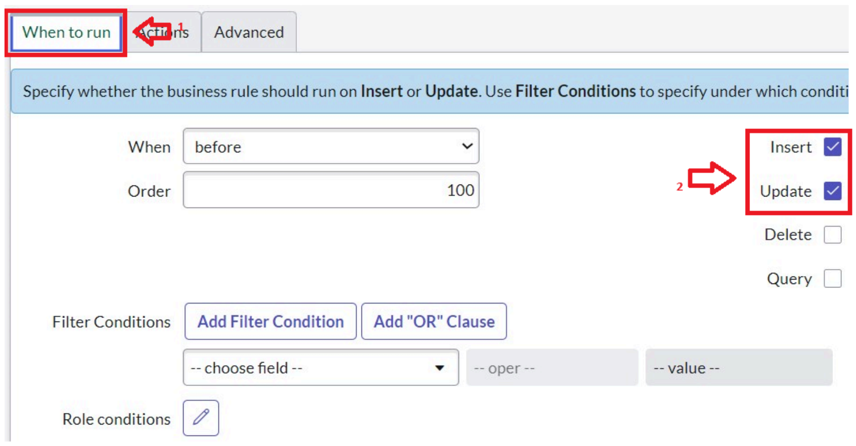
Check Advanced


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	Family Expenses BR	Application	Global
Table	Daily Expenses [u_daily_expenses]	Active	✓
		Advanced	✓

4. In when to run Check Insert and Update



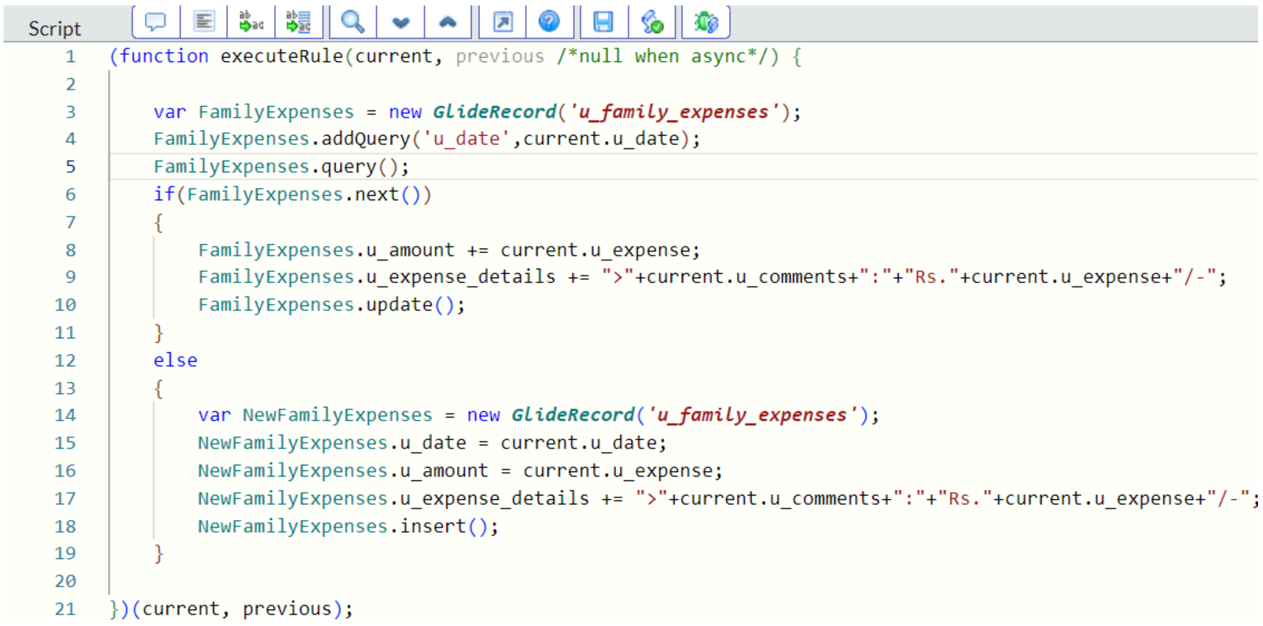
Specify whether the business rule should run on **Insert** or **Update**. Use **Filter Conditions** to specify under which conditi

When	before	Insert	✓
Order	100	Update	✓
		Delete	
		Query	

Filter Conditions: Add Filter Condition | Add "OR" Clause
-- 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);
```



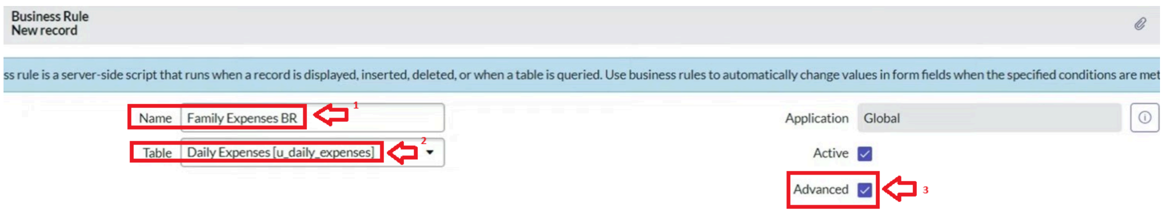
```
(function executeRule(current, previous /*null when async*/) {

    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);
```

6. Go to the Header and right click there>> 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:

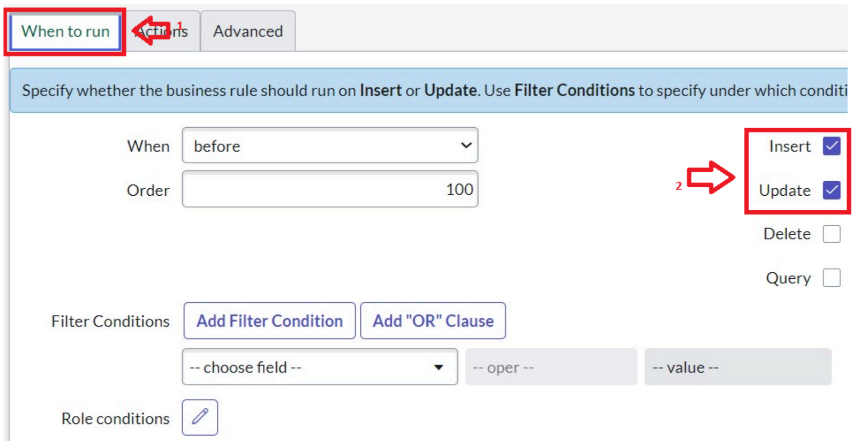
Name : Family Expenses BR
Table : Select Daily Expenses
Check Advanced


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	Family Expenses BR	Application	Global
Table	Daily Expenses [u_daily_expenses]	Active	✓
		Advanced	✓

4. In when to run Check Insert and Update



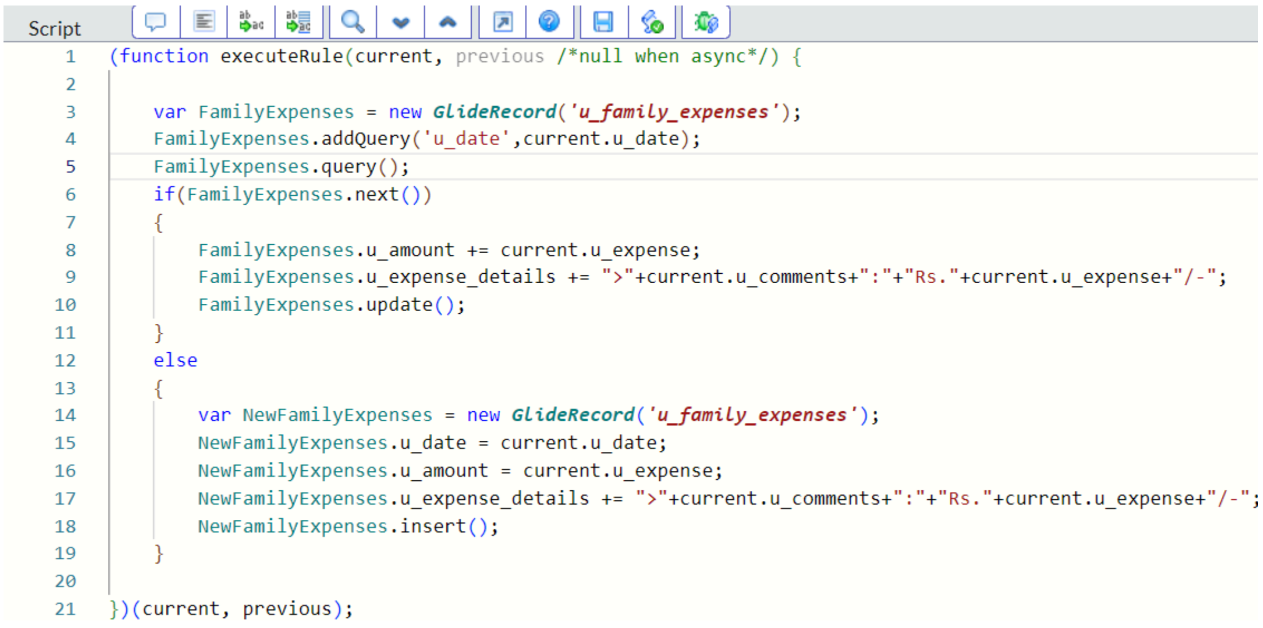
Specify whether the business rule should run on **Insert** or **Update**. Use **Filter Conditions** to specify under which conditi

When	before	Insert	✓
Order	100	Update	✓
		Delete	
		Query	

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);
```



```
(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);
```

6. Go to the Header and right click there>> click on Save.

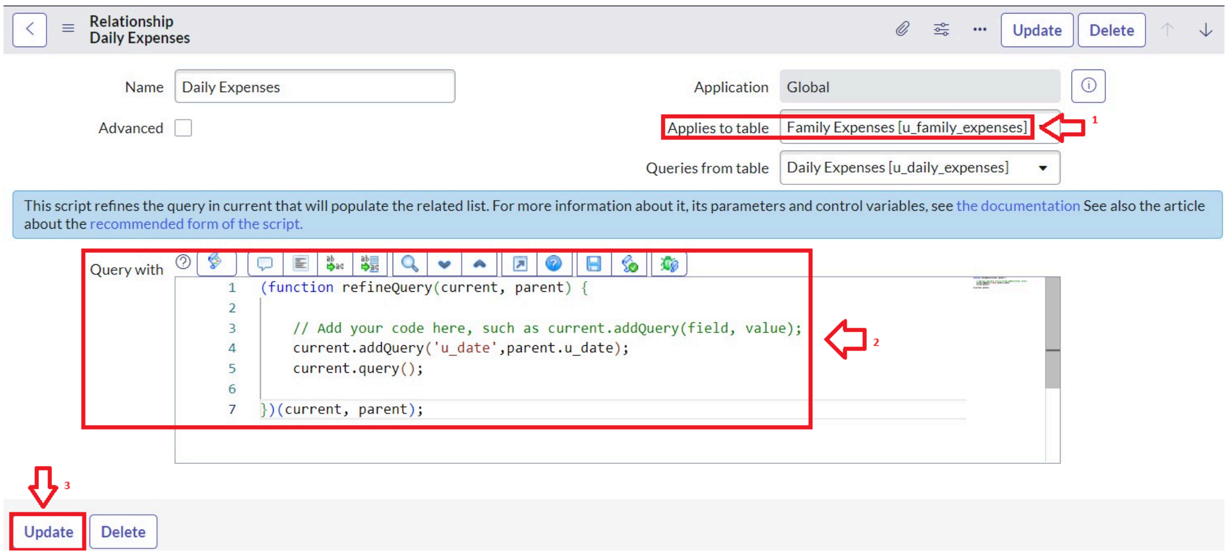
Configure The Relationship

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.



Name	Daily Expenses	Application	Global
Advanced	☐	Applies to table	Family Expenses [u_family_expenses]
		Queries from table	Daily Expenses [u_daily_expenses]

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

```
(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);
```

7. Results & Discussion:

The system was successfully implemented and tested.

Observed Results:

Family expenses are automatically updated as daily entries are made.

Reports show consolidated views of expenses for any given date.

Mandatory fields ensure data accuracy.

Relationships between tables allow families to drill down into daily records.

Advantages:

Simple to use, even for non-technical users.

Reduces manual work and errors.

Centralized system accessible from any device with internet access.

Extensible for future needs.

Conclusion & Future Enhancements

Future Enhancements:

Mobile app integration for quick expense entry.

Graphical dashboards (charts, pie charts, trend analysis).

AI-powered budgeting tips and spending alerts.

Multi-language support for broader usability.

Export and backup of data in Excel or PDF

9. References:

1. ServiceNow Developer Documentation –
https://developer.servicenow.com

2. GitHub Documentation – https://docs.github.com

3. Tutorials and online forums related to ServiceNow customization