Project Title: Calculation of Family Expenses using ServiceNow
Submitted by: Khadir Shaik
Role: ServiceNow Technical Consultant Intern Aspirant
Date: 18-09-2025
Mentor/Guide: Ajay

Abstract

This project, *“Calculation of Family Expenses using ServiceNow”*, aims to design and implement a comprehensive solution for managing and calculating family expenses. Leveraging the ServiceNow platform, the system streamlines the process of recording, categorizing, and aggregating daily expenses into a centralized Family Expenses table. The project demonstrates how automation through Business Rules ensures real-time updates and accuracy in financial tracking. This enables families to monitor spending effectively, make informed financial decisions, and maintain transparency in expense management.

Index

1. Introduction

In today’s world, families require effective methods to track and manage their daily and monthly expenses. Manual tracking often leads to errors, lack of visibility, and difficulty in consolidation. ServiceNow, as a cloud-based platform, provides robust automation and customization capabilities to build such solutions efficiently.

This project implements an **automated expense management system** on ServiceNow that categorizes expenses, tracks daily spending, and aggregates them into a consolidated family expense report.

2. Objectives

- To automate family expense calculation using ServiceNow.
- To provide a centralized system for recording and managing expenses.
- To enable automatic summation of expenses based on daily inputs.
- To ensure real-time visibility into financial data for informed decision-making.

3. Tools & Technologies Used

- **ServiceNow**: Core platform for building the application.
- **GlideRecord & Business Rules**: For backend automation.
- **Update Sets**: For managing project changes.
- **ServiceNow Form Design & Relationships**: For UI customization.

4. System Design & Architecture

The system consists of two core entities:

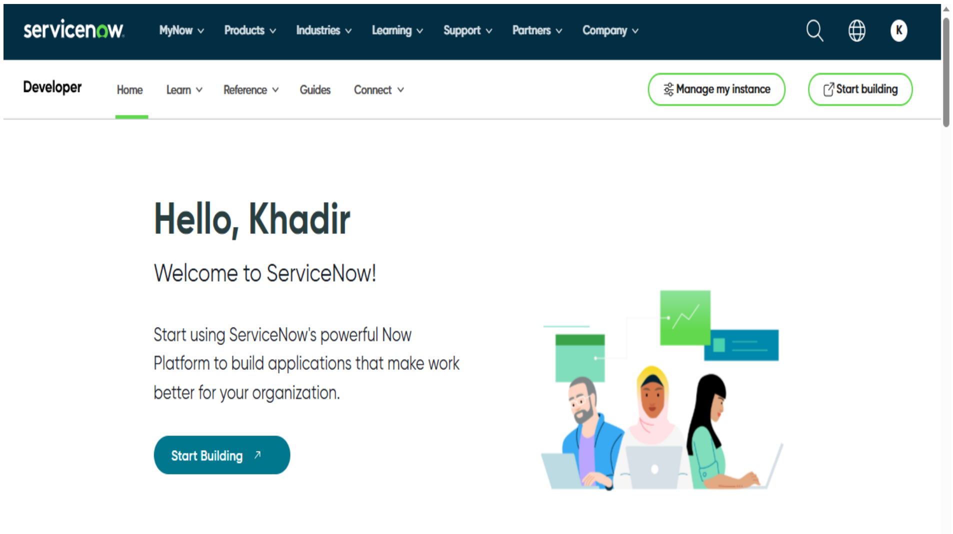
- **Family Expenses Table** – Aggregates total expenses for each date.
- **Daily Expenses Table** – Stores individual expense entries per family member.

A **Business Rule** automatically updates the Family Expenses table whenever a new Daily Expense is submitted.

5. Implementation Steps

5.1 ServiceNow Instance Setup

- Signed up for a **ServiceNow Developer Instance**.
- Requested and activated a **Personal Developer Instance (PDI)**.



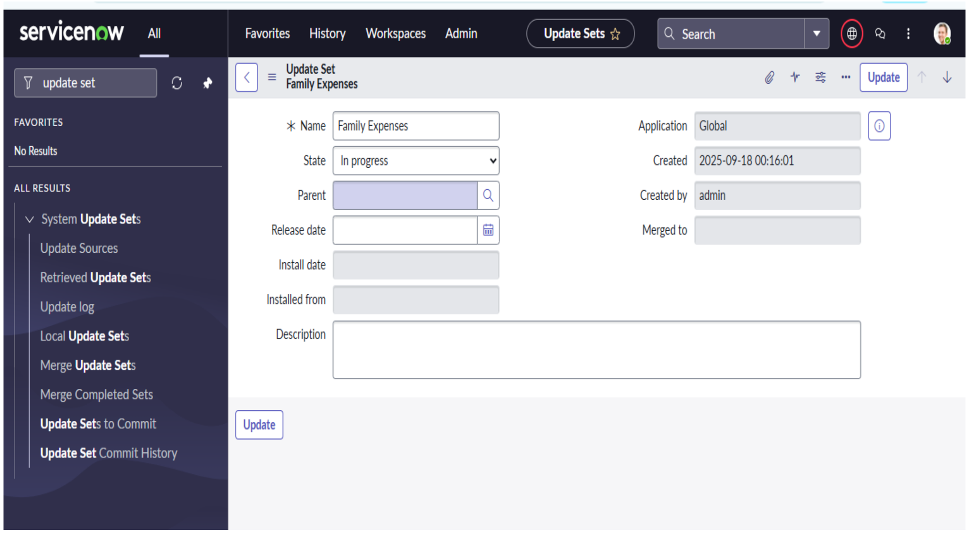
Hello, Khadir

Welcome to ServiceNow!

Start using ServiceNow's powerful Now Platform to build applications that make work better for your organization.

5.2 Creation of Update Set

- Created an update set named **“Family Expenses”** to track project changes.



Update Set
Family Expenses

* Name	Family Expenses	Application	Global
State	In progress	Created	2025-09-18 00:16:01
Parent		Created by	admin
Release date		Merged to	
Install date			
Installed from			
Description			

5.3 Table Creation

- **Family Expenses Table**: Contains *Number, Date, Amount, Expense Details*.
- **Daily Expenses Table**: Contains *Number, Date, Expense, Family Member Name, Comments*.

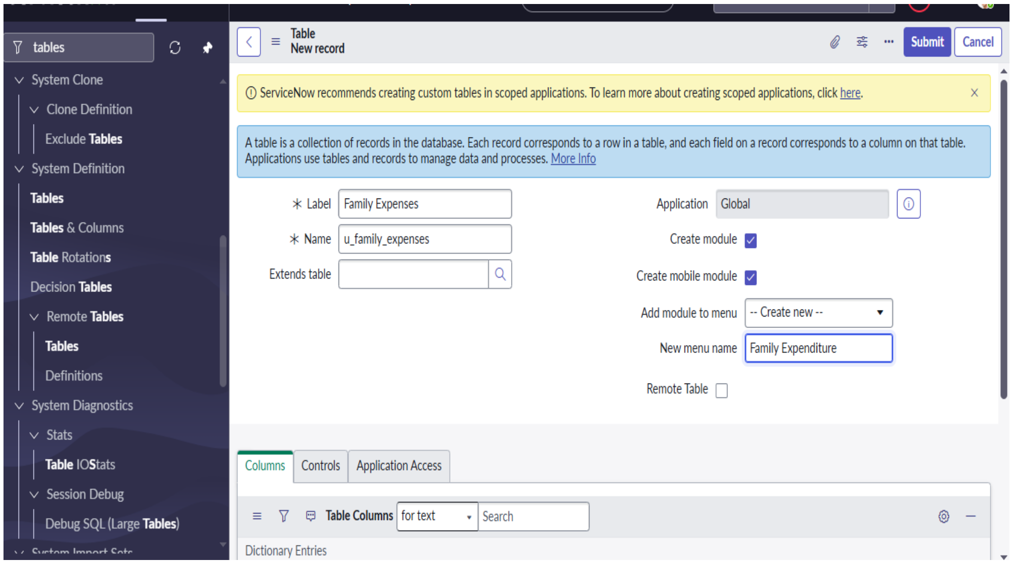


Table
New record

ⓘ ServiceNow recommends creating custom tables in scoped applications. To learn more about creating scoped applications, click here.

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	☑
Extends table		Create mobile module	☑
		Add module to menu	-- Create new --
		New menu name	Family Expenditure
		Remote Table	□

Table Columns for text

Dictionary Entries

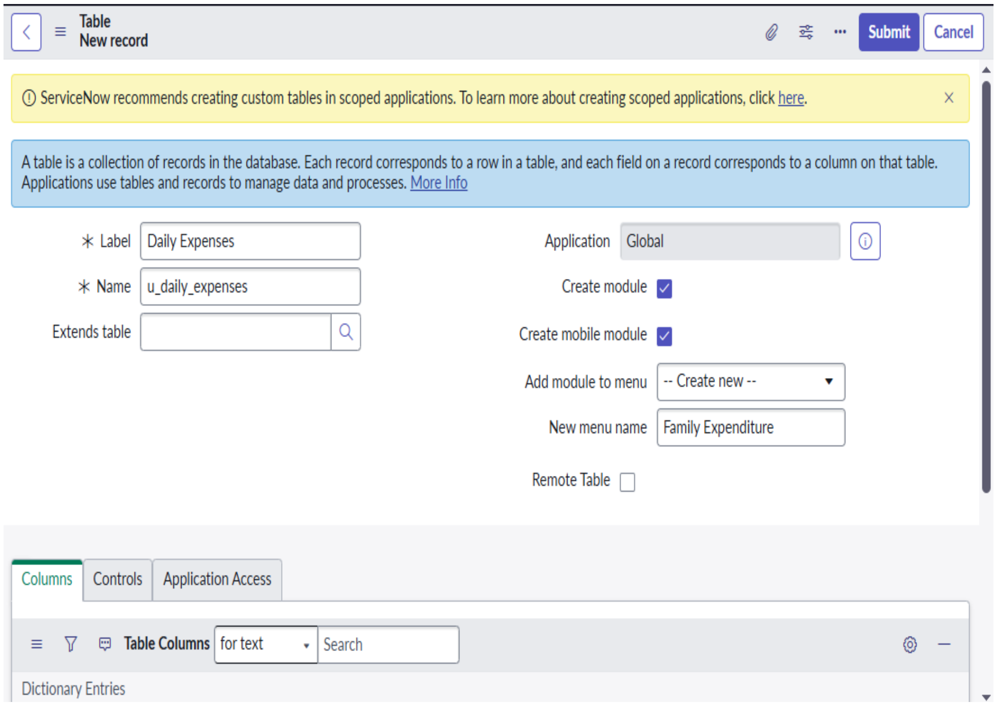


Table
New record

ⓘ ServiceNow recommends creating custom tables in scoped applications. To learn more about creating scoped applications, click here.

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	Create module	☑
Extends table		Create mobile module	☑
		Add module to menu	-- Create new --
		New menu name	Family Expenditure
		Remote Table	□

Table Columns for text

Dictionary Entries

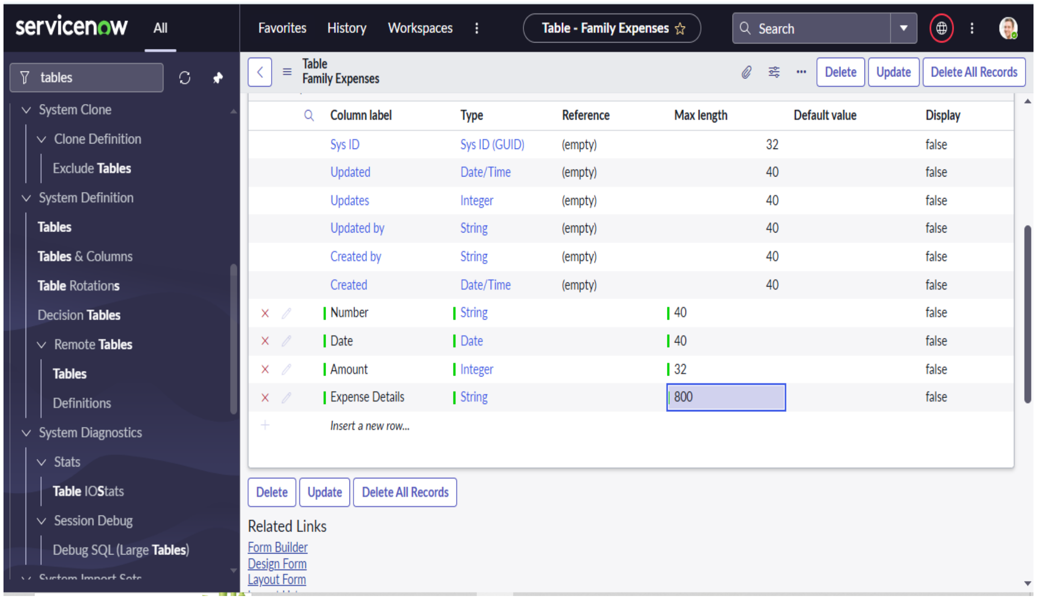


Table
Family Expenses

Column label	Type	Reference	Max length	Default value	Display
Sys ID	Sys ID (GUID)	(empty)	32		false
Updated	Date/Time	(empty)	40		false
Updates	Integer	(empty)	40		false
Updated by	String	(empty)	40		false
Created by	String	(empty)	40		false
Created	Date/Time	(empty)	40		false
Number	String		40		false
Date	Date		40		false
Amount	Integer		32		false
Expense Details	String		800		false
Insert a new row...					

Related Links
Form Builder
Design Form
Layout Form

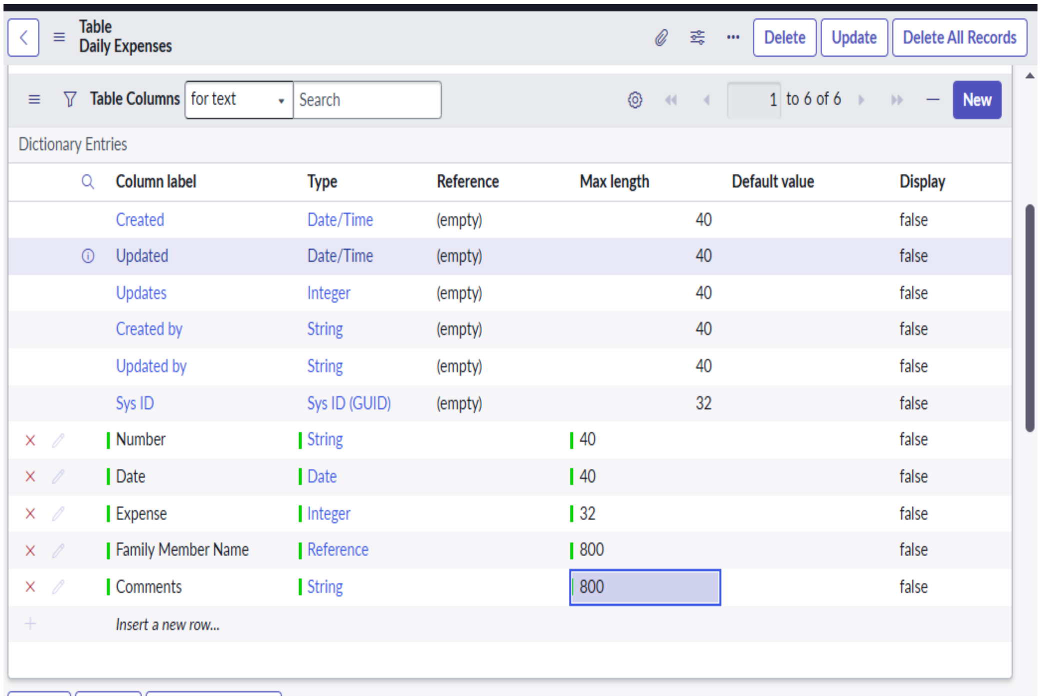


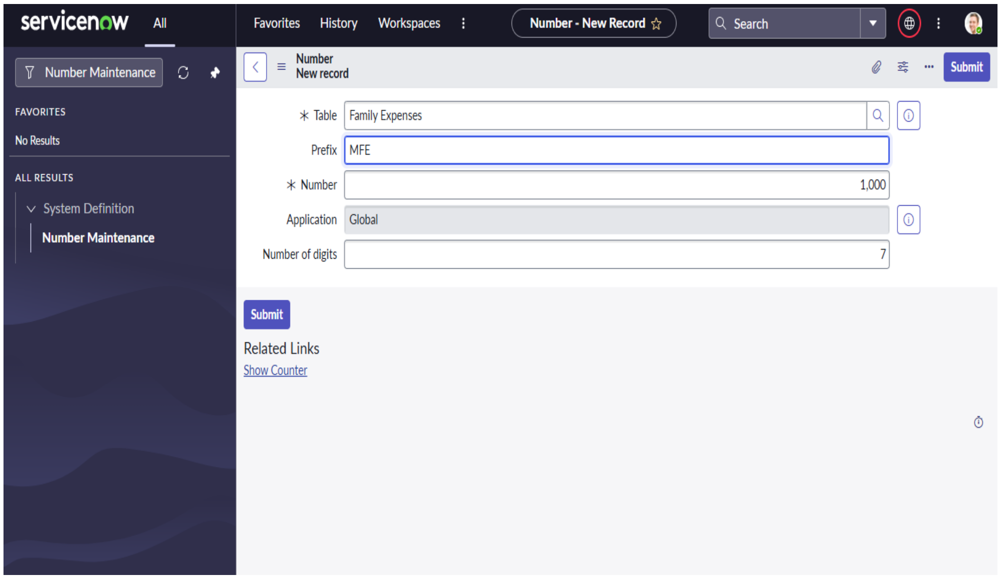
Table
Daily Expenses

Dictionary Entries

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

5.4 Form Configuration

- Number field → Auto-generated & Read-only.
- Mandatory fields: Date, Amount, and Family Member Name.
- Related List: Daily Expenses mapped under Family Expenses.

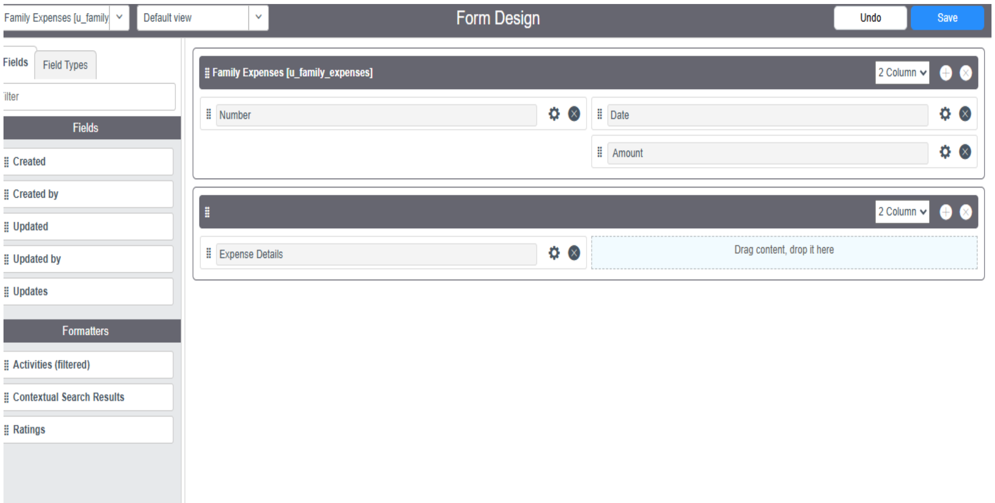


* Table	Family Expenses
Prefix	MFE
* Number	1,000
Application	Global
Number of digits	7

Submit

Related Links

Show Counter



Family Expenses [u_family_expenses] — 2 Column

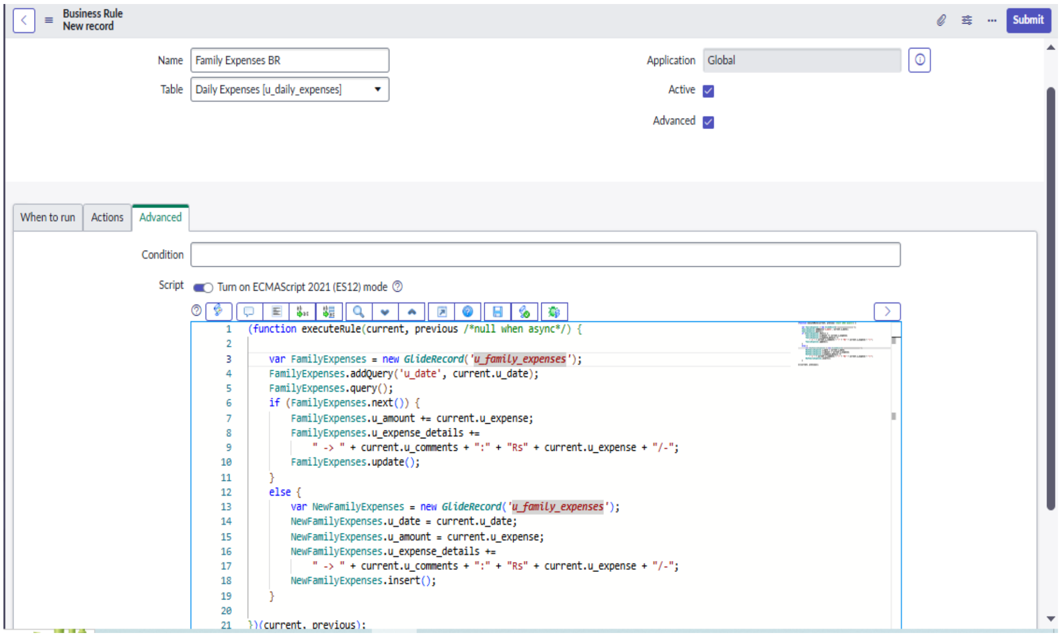
Number	Date
	Amount

2 Column

Expense Details	Drag content, drop it here

5.5 Business Rules Implementation

A **Business Rule** was created to auto-update Family Expenses whenever a Daily Expense record is added or updated.



Name: Family Expenses BR

Table: Daily Expenses [u_daily_expenses]

Application: Global

Active ☑

Advanced ☑

Condition:

Script: Turn on ECMAScript 2021 (ES12) mode

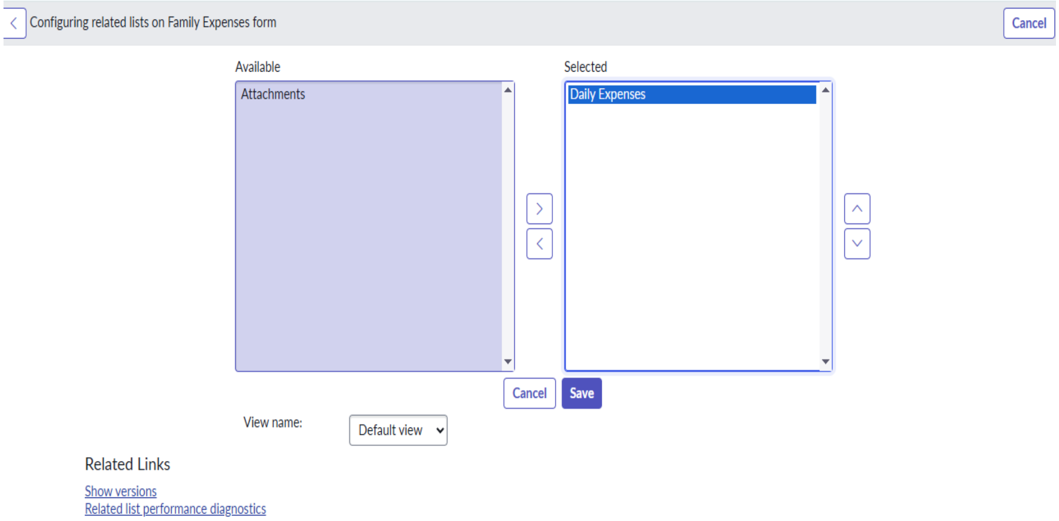
```
(function executeRule(current, previous /*null when async*/) {

    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);
```

Code Snippet:

```
(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);
```

5.6 Relationships & Related Lists

- Configured **Daily Expenses** → **Family Expenses** relationship.
- Added **Daily Expenses as a Related List** under Family Expenses for clear visibility.



Configuring related lists on Family Expenses form

Available: Attachments

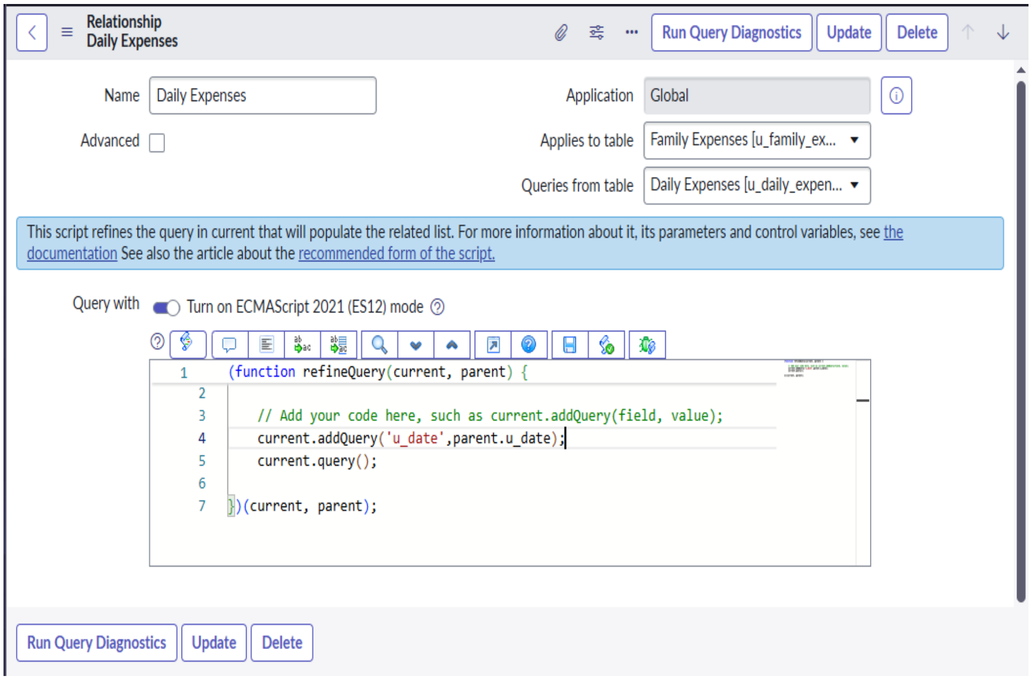
Selected: Daily Expenses

View name: Default view

Related Links

Show versions

Related list performance diagnostics



Relationship
Daily Expenses

Name: Daily Expenses

Advanced: □

Application: Global

Applies to table: Family Expenses [u_family_ex...

Queries from table: Daily Expenses [u_daily_expen...

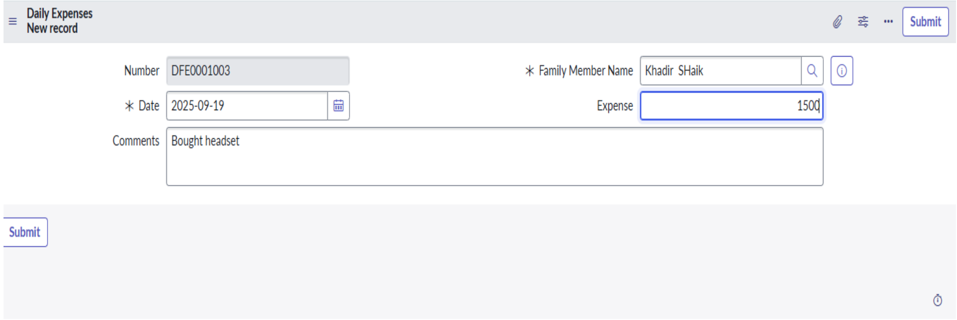
This script refines the query in current that will populate the related list. For more information about it, its parameters and control variables, see the documentation See also the article about the recommended form of the script.

Query with: Turn on ECMAScript 2021 (ES12) mode

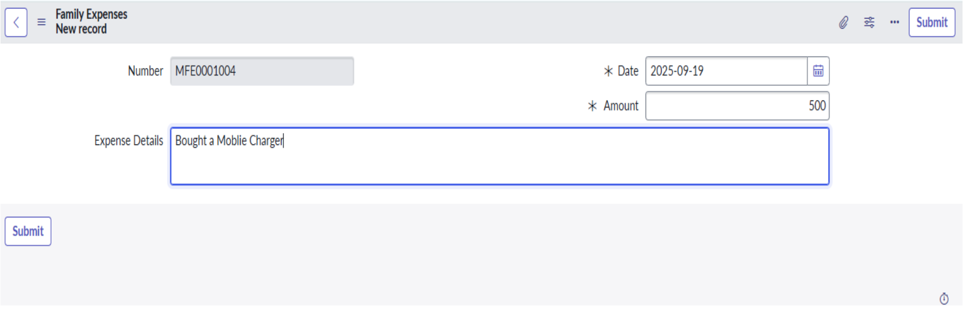
```
(function refineQuery(current, parent) {

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

})(current, parent);
```

6. Testing & Results

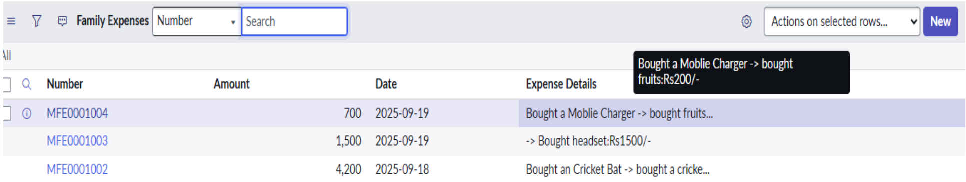
- Inserted multiple rows into **Daily Expenses**.
- Verified that the **Family Expenses table automatically updated** with total amount and consolidated comments.



Number	DFE0001003	* Family Member Name	Khadir SHaik
* Date	2025-09-19	Expense	1500
Comments	Bought headset		

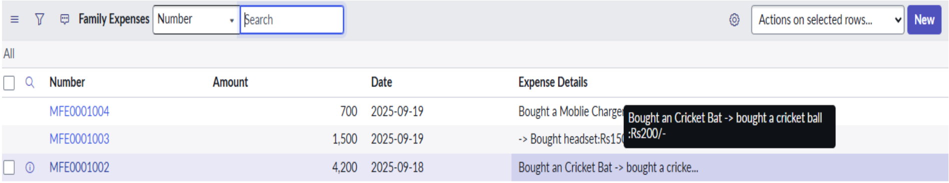


Number	MFE0001004	* Date	2025-09-19
		* Amount	500
Expense Details	Bought a Moblie Charger		



Bought a Moblie Charger -> bought fruits:Rs200/-

Number	Amount	Date	Expense Details
MFE0001004	700	2025-09-19	Bought a Moblie Charger -> bought fruits...
MFE0001003	1,500	2025-09-19	-> Bought headset:Rs1500/-
MFE0001002	4,200	2025-09-18	Bought an Cricket Bat -> bought a cricke...



Number	Amount	Date	Expense Details
MFE0001004	700	2025-09-19	Bought a Moblie Charge
MFE0001003	1,500	2025-09-19	-> Bought headset:Rs15
MFE0001002	4,200	2025-09-18	Bought an Cricket Bat -> bought a cricke...

Bought an Cricket Bat -> bought a cricket ball :Rs200/-

7. Benefits of the System

- Eliminates manual expense tracking errors.
- Provides real-time aggregated family spending insights.
- Supports scalability for different family sizes and expense categories.
- Enhances financial transparency and decision-making.

8. Conclusion

The *Calculation of Family Expenses using ServiceNow* project successfully demonstrates how ServiceNow can be leveraged to automate household financial tracking. Through **tables, business rules, and relationships**, the system ensures accurate, real-time expense aggregation. This not only improves efficiency but also highlights ServiceNow's adaptability beyond enterprise use cases.

9. References

- **Project Repository**
  - Family Expense Management using ServiceNow → GitHub Repository
- **ServiceNow Official Documentation**
  - ServiceNow Documentation Home
  - ServiceNow Tables Overview
  - Business Rules in ServiceNow
  - Update Sets in ServiceNow
  - Flow Designer Overview
  - Relationships in ServiceNow
- **Learning Resources**
  - ServiceNow Developer Portal
  - GlideRecord API Reference