



**SHREE CHANDRAPRABHU  
JAIN COLLEGE - MINJUR**  
(Affiliated to the University of Madras)  
(Accredited by NAAC)



## **Project Title: CALCULATING FAMILY EXPENSES USING SERVICENOW**

### **Team Id:**

*Team Members: 4*

Team Leader: B.GOPIKA

Team Member 1:G.PRINCY

Team Member 2:M.YABEZH

Team Member 3:B.VISHNU

**Problem Statement:** Daily and monthly expenses are hard to manage manually, so an automated solution is required.

**Objective:** To develop a small ServiceNow app for recording daily expenses and auto-calculating monthly totals.

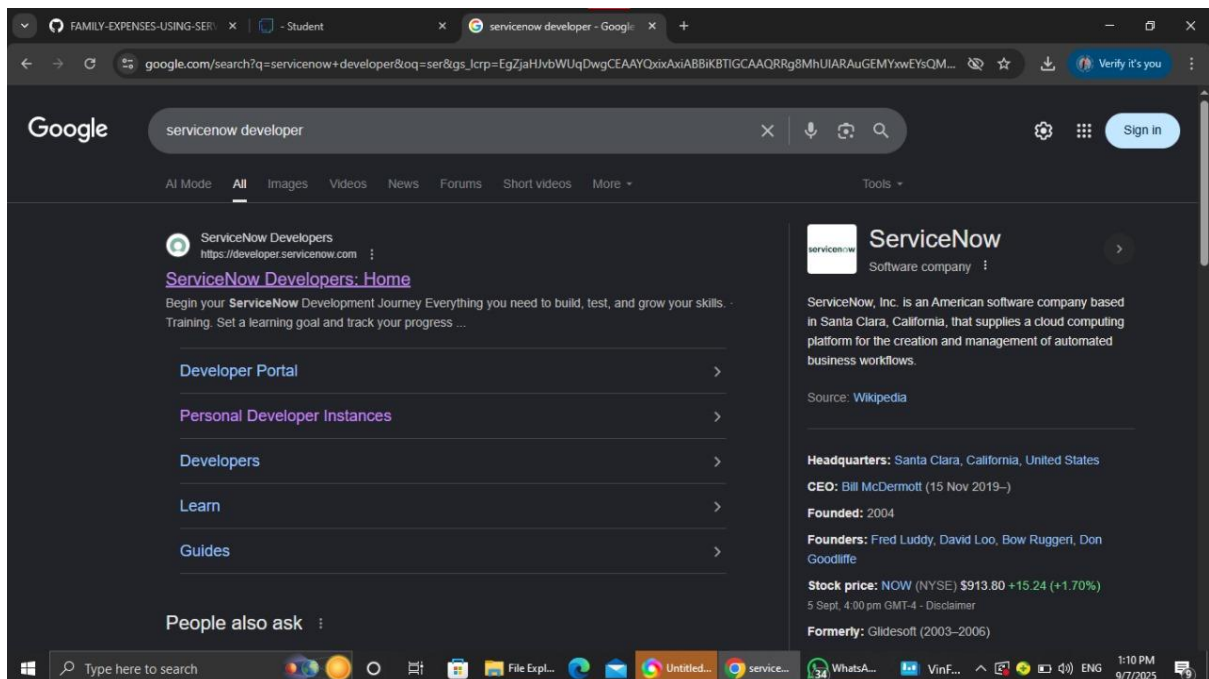
**Skills:** ServiceNow basics, JavaScript, reporting.

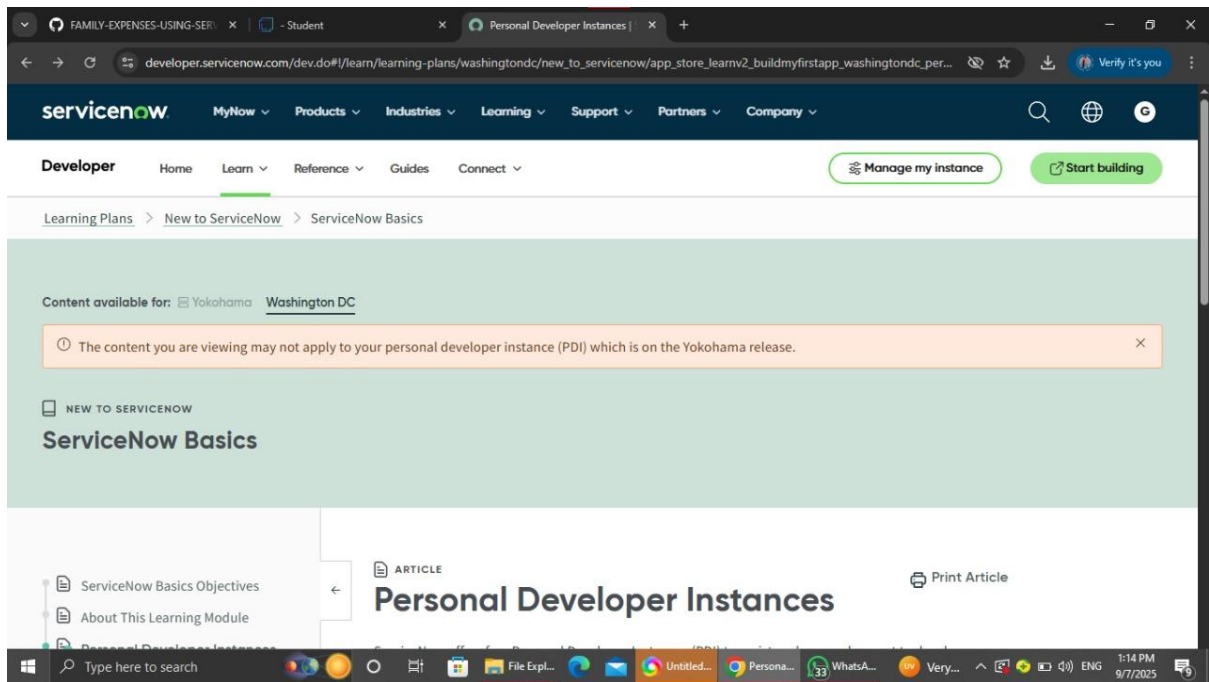
### **TASK INITIATION**

#### **Milestone 1:**

**Activity 1: 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.

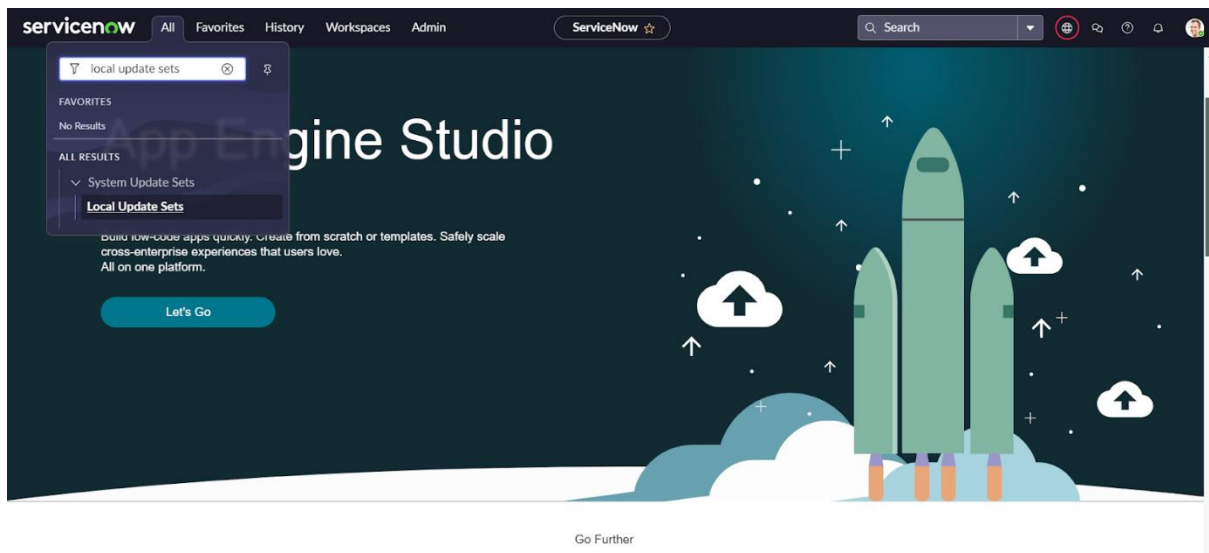




## Milestone 2:

### Activity 2: Creation of New Update Set

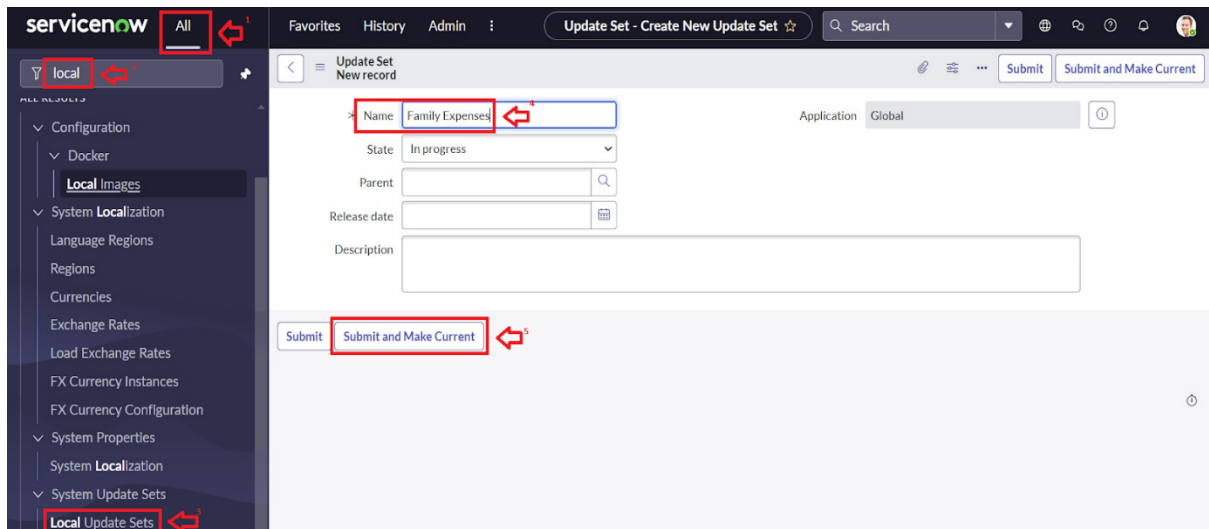
1. Go to All >> In the filter search for Local Update set > click on New.



2. Enter the Details as:

Name : Family Expenses

3. Then click on Submit and Make current.



### Milestone 3: Creation of tables

#### Activity 1: 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

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.

#### Activity2: 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

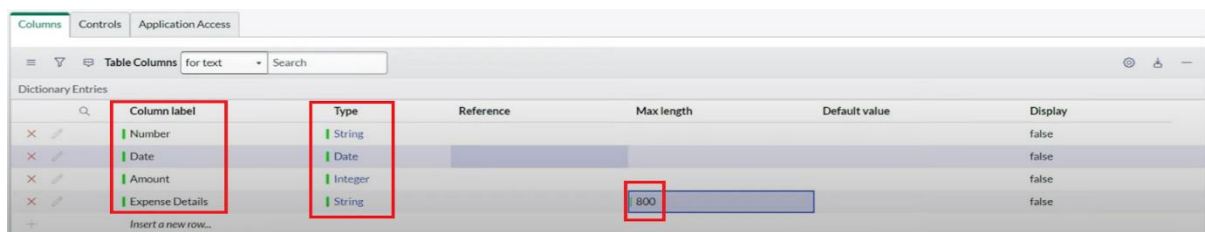
7. Double click on insert a new row again

8. Give the details as:

Column label : Expense Details

Type : String

Max length : 800



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

### Activity3: 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

< | Number MFE | Update | Delete

\* Table: Family Expenses

Prefix: MFE

\* Number: 1,000

Application: Global

Number of digits: 7

Update | Delete

9. Click on Submit.

#### Activity4: 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.

#### Milestone4:

##### Activity1: 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

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

### **Activity2: 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:

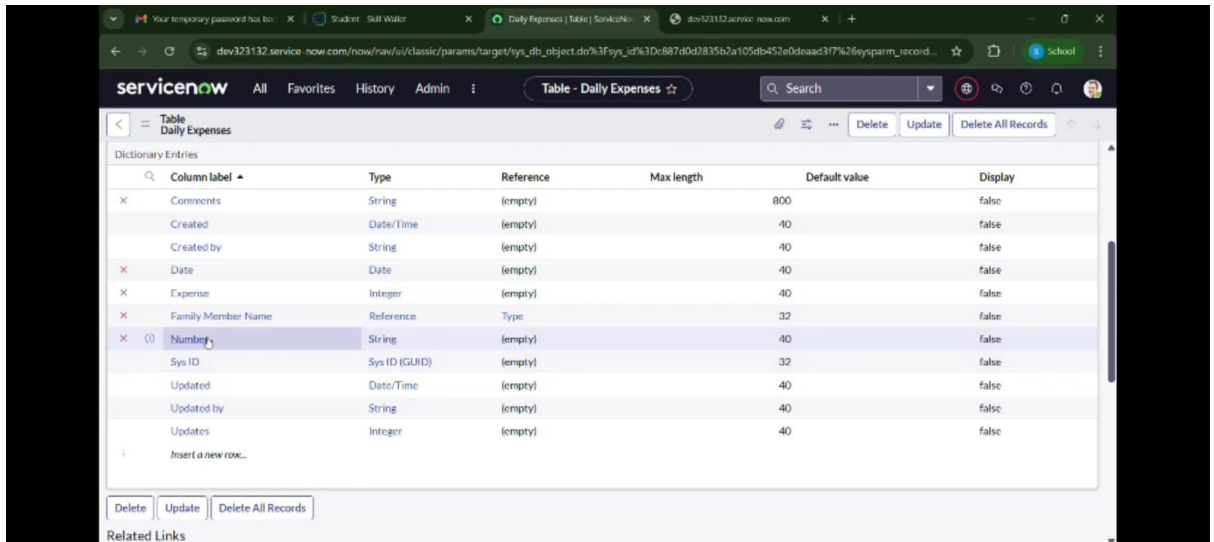
Column label : Comments

Type : String

Max length : 800



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



- 12.

### Activity3: 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.



- 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 : Daily Expenses

Prefix : DFE

The screenshot shows a 'New record' form for 'Number Maintenance'. The form has a header bar with a back arrow, a menu icon, the text 'Number New record', a help icon, a settings icon, and a 'Submit' button. The form fields are: '\* Table' with the value 'Daily Expenses' (annotated with a red box and arrow 1), '\* Prefix' with the value 'DFE' (annotated with a red box and arrow 2), '\* Number' with a value field and '1,000' on the right, 'Application' with the value 'Global' (annotated with a red box and arrow 3), and 'Number of digits' with a value field and '7' on the right. A 'Submit' button is at the bottom left, also annotated with a red box and arrow 3.

9. Click on Submit.

#### Activity4: 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.

The screenshot shows a form interface for 'Daily Expenses [u\_daily\_expenses]'. The form has a header bar with the title and a '2 Column' dropdown. Below the header, there are four input fields arranged in a 2x2 grid: 'Number', 'Date', 'Family Member Name', and 'Expense'. Each field has a gear icon to its right, indicating a configuration menu. Below this grid is a 'Comments' section with a single text input field and a gear icon.

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.

### **Milestone5:**

#### **Activity1: 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.

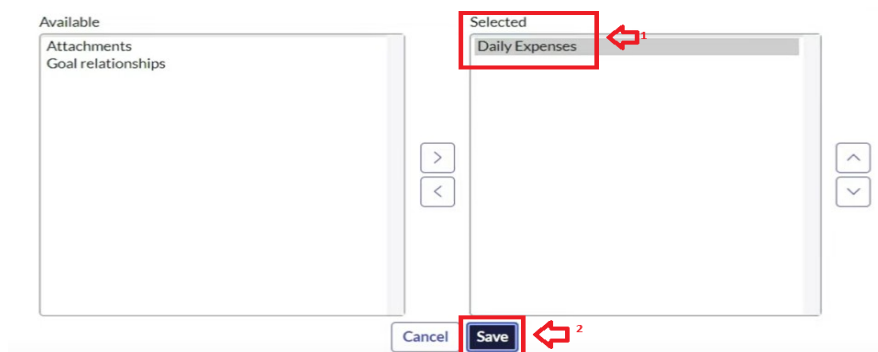
### **Milestone6:**

#### **Activity1: 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



## Milestone7:

### Activity1: 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

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.

Name: Family Expenses BR ← 1

Table: Daily Expenses [u\_daily\_expenses] ← 2

Application: Global 0

Active ☒

Advanced ☒ ← 3

#### 4. In when to run Check Insert and Update

When to run ← 1 Advanced

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

When: before

Order: 100

Filter Conditions: Add Filter Condition Add "OR" Clause

-- choose field -- -- oper -- -- value --

Role conditions: ✎

Insert ☒ ← 2

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);
```

```

Script
1 (function executeRule(current, previous /*null when async*/) {
2
3     var FamilyExpenses = new GlideRecord('u_family_expenses');
4     FamilyExpenses.addQuery('u_date', current.u_date);
5     FamilyExpenses.query();
6     if(FamilyExpenses.next())
7     {
8         FamilyExpenses.u_amount += current.u_expense;
9         FamilyExpenses.u_expense_details += ">" + current.u_comments + ":" + "Rs." + current.u_expense + "/-";
10        FamilyExpenses.update();
11    }
12    else
13    {
14        var NewFamilyExpenses = new GlideRecord('u_family_expenses');
15        NewFamilyExpenses.u_date = current.u_date;
16        NewFamilyExpenses.u_amount = current.u_expense;
17        NewFamilyExpenses.u_expense_details += ">" + current.u_comments + ":" + "Rs." + current.u_expense + "/-";
18        NewFamilyExpenses.insert();
19    }
20
21 }(current, previous);

```

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

## Milestone8:

### Activity1:

#### 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.

The screenshot shows a web form titled "Relationship Daily Expenses". At the top right are "Update" and "Delete" buttons. Below the title, there are fields for "Name" (containing "Daily Expenses"), "Application" (set to "Global"), and "Advanced" (unchecked). A red box highlights the "Applies to table" field, which contains "Family Expenses [u\_family\_expenses]"; a red arrow labeled "1" points to this field. Below it, the "Queries from table" dropdown is set to "Daily Expenses [u\_daily\_expenses]". A blue informational banner states: "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](#)." Below the banner is a "Query with" section containing a code editor. A red box highlights the code editor, and a red arrow labeled "2" points to it. The code in the editor is:

```
1 (function refineQuery(current, parent) {  
2  
3     // Add your code here, such as current.addQuery(field, value);  
4     current.addQuery('u_date', parent.u_date);  
5     current.query();  
6  
7 })(current, parent);
```

At the bottom left, a red arrow labeled "3" points to the "Update" button.

Relationship  
Daily Expenses

Name

Application

Advanced ☐

Applies to table  <sup>1</sup>

Queries from table

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

```
1 (function refineQuery(current, parent) {  
2  
3     // Add your code here, such as current.addQuery(field, value);  
4     current.addQuery('u_date', parent.u_date);  
5     current.query();  
6  
7 })(current, parent);
```

<sup>2</sup>

<sup>3</sup>

Update Delete