

PERFORMANCE & TESTING

Procedure or Implementation steps

Phase 1 : 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

Phase 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.

Phase 3 : Creation of Table

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

Step 2: 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

Step 3: 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.

Phase 4 : Creation of Relationship between Family Expenses and Daily Expenses tables

1. Click on all >> search for service catalog
2. Select maintain item under catalog definition
3. Search for ‘laptop request’ which is created before
4. Select ‘laptop request’ and scroll down click on “Catalog Ui policies”
5. In the catalog ui policies related list tab click on new
6. Give short description as: show accessories details

Phase 6 : 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

Phase 7 : 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

Phase 8 : 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.

Screenshots:

Phase 1 : Creation of New Update Set

The screenshot shows the 'Update Set - Family Expenses' page in ServiceNow. The main form contains fields for Name (Family Expenses), State (In progress), Parent (empty), Release date, Install date, Installed from, and Description. To the right, details like Application (Global), Created (2025-10-29 01:33:07), Created by (admin), and Merged to are listed. Below the form is a Related Links section with 'Merge With Another Update Set' and 'Scan Update Set'. A navigation bar at the bottom includes Customer Updates (43), Update Set Logs, Child Update Sets, and Install History. A table below shows the created update set with columns for Created, Type, View, Target name, Updated by, Remote update set, and Action.

Created	Type	View	Target name	Updated by	Remote update set	Action
2025-10-29 01:37:03	Application Menu		Family Expenditure	admin	(empty)	INSERT_OR_UPDATE

Phase 2 :

1. Creation of Family Expenses Table

The screenshot shows the 'Table - New Record' page for creating a new table named 'Family expenses'. The form includes fields for Label (Family expenses), Name (u_family_expenses), and Extends table (empty). Configuration options include Application (Global), Create module (checked), Create mobile module (checked), Add module to menu (Create new), and New menu name (Family Expenditure). Below the form is a table for Dictionary Entries with columns for Column label, Type, Reference, Max length, Default value, and Display. The table has a single row with a plus sign to add a new row.

Column label	Type	Reference	Max length	Default value	Display
+					

2 . Create a columns

The screenshot shows the 'Columns' tab in the ServiceNow interface. The table has the following data:

Column label	Type	Reference	Max length	Default value ▾	Display
Number	String				false
Date	Date				false
Expense Details	String		800		false
Amount	Integer				false

3 . Making Number Field an Auto-Number

The screenshot shows the 'Number - New Record' screen. The configuration fields are:

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

4. Configure the Form



Phase 3 : Creation of Relationship between Family Expenses and Daily Expenses tables

Name: Daily Expenses

Application: Global

Advanced:

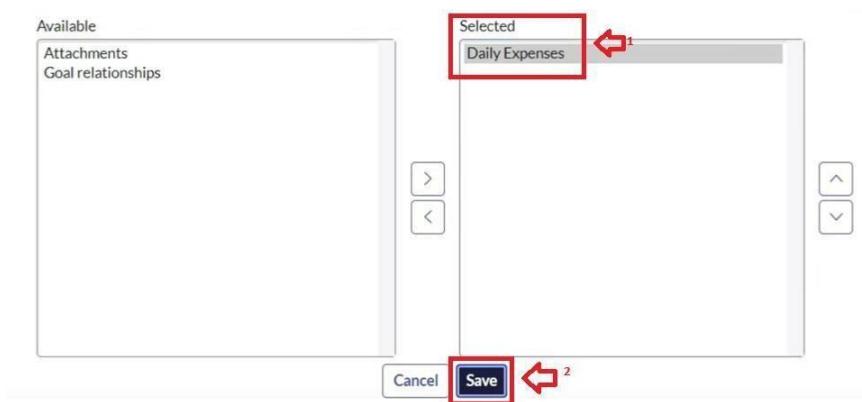
Applies to table: Family Expenses [u_st_family_expenses]

Queries from table: -- None --

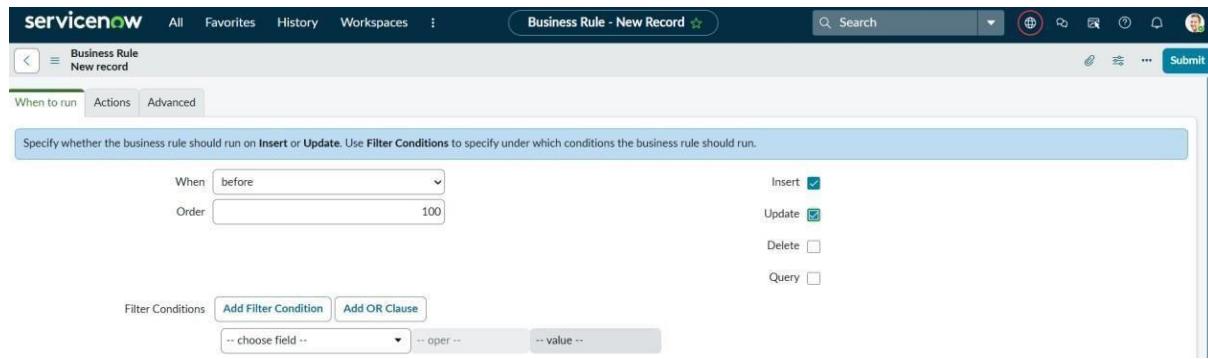
```
1 (function refineQuery(current, parent) {  
2     // Add your code here, such as current.addQuery(field, value);  
3  
4 })(current, parent);  
5 )
```

Submit

Phase 4: Configuring Related List on Family Expenses



Phase 5: Creation of Business Rules



```
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);
```

Phase 6: Configure the Relationship

The screenshot shows the ServiceNow interface for creating a new relationship record. The top navigation bar includes 'servicenow', 'All', 'Favorites', 'History', 'Workspaces', and a search bar. The main title is 'Relationship - New Record'. The form has a 'Name' field set to 'Daily Expenses', an 'Application' dropdown set to 'Global', and an 'Applies to table' dropdown set to 'Family Expenses (u_st_family_expenses)'. Below these are 'Advanced' and 'Queries from table' settings. A note at the bottom 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.' A code editor window contains the following JavaScript:

```
Query with: Turn on ECMAScript 2021 (ES12) mode
1 (function refineQuery(current, parent) {
2
3     // Add your code here, such as current.addQuery(field, value);
4
5 })(current, parent);
```

A 'Submit' button is located at the bottom left of the form.