

Date	25 June 2025
Team ID	LTVIP2025TMID20412
Project Name	Calculating Family Expenses using Service Now
Mentor Name	Dr Shaik Salma Begum
Maximum Marks	10 Marks

Calculating Family Expenses using Service Now

Milestone 1: Setting Up ServiceNow Instance:

- Sign up on developer.servicenow.com and create a developer account.
- Navigate to the Personal Developer Instance section and request a new instance.
- Fill out the required information and submit the request.
- Instance details (URL, username, password) will be sent via email.
- Log in to the instance using the provided credentials.
- The instance is ready for development and customization.

Milestone 2: Creation of New Update Set:

- Creating a Local Update Set in ServiceNow
 - Navigate to All in the application navigator.
 - In the filter search bar, type Local Update Set and select it from the results.
 - Click on the New button to create a new update set.
 - Name: Family Expenses
 - In the form that appears, enter the following details:
 - Click on Submit to save the update set.
 - After submission, click on Make Current to set this update set as the active one.
- Then created update set called “Family Expenses” and made it current

The screenshot shows the ServiceNow interface for creating a new update set. The top navigation bar includes the ServiceNow logo, navigation tabs (All, Favorites, History, Workspaces), and a search bar. The main header is 'Update Set - Create New Update Set'. Below this, there is a sub-header 'Update Set New record' and a 'Submit' button. The form fields are as follows:

- Name:** Family Expenses
- Application:** Global
- State:** In progress
- Parent:** (empty)
- Release date:** (empty)
- Description:** (empty)

At the bottom of the form, there are two buttons: 'Submit' and 'Submit and Make Current'.

Milestone 3: Creation of Table (Family Expenses):

- Creation of Family Expenses tables and its fields:

Family Expenses Table Fields

Field Name	Type	
Number	String	Auto populate Number with Prefix MFE
Date	Date	
Amount	Integer	
Expense Details	String	Max Length 800

- Navigate to All in the application navigator.
- In the filter search bar, type Tables and select it from the results.
- Click on the New button to create a new table.
- In the form, provide the following details:
 - Label: Family Expenses
 - Name: Auto-Populated (automatically generated based on the label)
 - New menu name: Family Expenditure
- Navigate to the header section of the form, right-click on the header, and select Save from the context menu.

The screenshot shows the ServiceNow interface for configuring the 'Family Expenses' table. The 'Columns' tab is selected, displaying a list of dictionary entries. The 'Number' field is highlighted, indicating it is the current selection for configuration.

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

- Making Number Field an Auto-Number:
 - In the Columns section of the *Family Expenses* table, locate and double-click on the Number field to open its configuration.
 - Scroll down and double-click on Advanced view to expand additional settings.
 - Under the Default Value section:
 - Enable the checkbox for Use dynamic default.
 - Set the Dynamic default value to Get Next Padded Number.

- Click on Update to save the changes to the field.

The screenshot shows the 'Dictionary Entry - Number' configuration page in ServiceNow. The 'Table' is set to 'Family Expenses [u_family_expenses]'. The 'Type' is 'String', 'Column label' is 'Number', 'Column name' is 'u_number', and 'Max length' is '40'. The 'Application' is 'Global'. The 'Active' checkbox is checked. The 'Function field', 'Read only', 'Mandatory', and 'Display' checkboxes are unchecked. There is a section for 'Attributes' and a 'Default Value' section with a 'Dynamic default' checkbox checked and a 'Dynamic default value' of 'Get Next Padded Number'. The 'Update' button is at the bottom right.

- Setting Up Number Maintenance for the Family Expenses Table
 - Navigate to All in the application navigator.
 - In the filter search bar, type Number Maintenance and select it from the list.
 - Click on the New button to create a new number maintenance record.
 - Provide the following details:
 - Table: Family Expenses
 - Prefix: MFE
 - Click on Submit to save the configuration.

The screenshot shows the 'Number - New Record' form in ServiceNow. The 'Table' is 'Family Expenses', 'Prefix' is 'MFE', 'Number' is '1,000', 'Application' is 'Global', and 'Number of digits' is '7'. The 'Submit' button is at the bottom left.

- Configure the Form:
 - Navigate to All in the application navigator.
 - In the filter search bar, type Family Expenses and select it from the results to open the module.
 - Click on the New button to open a new record form.
 - In the form header, right-click and select Configure > Form Design from the dropdown menu.
 - In the Form Designer, use drag-and-drop functionality to customize the layout of the form fields as needed. Fields can be rearranged, moved between sections, or grouped based on specific requirements.
 - Once the desired layout has been achieved, click Save or Close to apply the changes.
 - Before Form design:

servicenow All Favorites History Workspaces **Family Expenses - Create MFE0001001** Search

Family Expenses
New record

Number MFE0001001

Expense Details

Date Amount

Submit

Form design:

Family Expenses [u_family] Default view Form Design

Fields Field Types

Filter

Fields

- Created
- Created by
- Updated
- Updated by
- Updated on

Formatters

- Activities (Filtered)
- Contextual Search Results
- Rating

Family Expenses [u_family_expenses] 2 Column

- Number
- Date
- Amount

Expense Details 1 Column

After form design:

servicenow All Favorites History Workspaces **Family Expenses - Create MFE0001002** Search

Family Expenses
New record

Number MFE0001002

* Date

* Amount

Expense Details

Submit

Milestone 4: Creation of Table (Daily Expenses):

- Creation of Daily Expenses Table and its fields:

Daily Expenses Table Fields

Field Name	Type	
Number	String	Auto populate Number with Prefix DFE
Family Member Name	Reference	Sys_User
Date	Date	
Expense	Integer	
Comments	String	Max Length 800

- Navigate to All in the application navigator.
- In the filter search bar, type Tables and select it from the list.
- Click on the New button to create a new table.
- Fill in the following details in the form:
 - Label: Daily Expenses
 - Name: Auto-Populated (automatically generated based on the label)
 - Add Module to menu: Family Expenditure
- Navigate to the header of the form, right-click, and select Save from the context menu.

Column label	Type	Reference	Max length	Default value	Display
Number	String				false
Date	Date				false
Expense	Integer				false
Family Member Name	Reference		800		false
Comments	String		800		false

- Making Number Field an Auto-Number:
 - In the Columns section of the Daily Expenses table, locate and double-click on the Number field to open its configuration.
 - Scroll down and double-click on Advanced view to expand additional settings.
 - Under the Default Value section:
 - Enable the checkbox for Use dynamic default.
 - Set the Dynamic default value to Get Next Padded Number.
 - Click on Update to save the changes to the field.

The screenshot shows the 'Dictionary Entry - Number' form in ServiceNow. The form is titled 'Dictionary Entry - Number' and has a 'View: Advanced' dropdown. A blue informational banner at the top explains that dictionary entries manage how ServiceNow stores data in tables and fields (columns). It instructs users to select a Table and the field Type of the new column, enter a column label, and the column name. It also mentions setting a Max length for text String type fields, making the field Mandatory to save a record, and making the field a Display Value for reference fields.

The form fields are as follows:

- * Table:
- * Type:
- * Column label:
- * Column name:
- * Max length:
- Application:
- Active: ☒
- Function field: ☐
- Read only: ☐
- Mandatory: ☐
- Display: ☐

Below the form fields is a section titled 'Attributes' with a text input field.

At the bottom, there are tabs for 'Choice List Specification', 'Calculated Value', and 'Default Value'. The 'Default Value' tab is selected. A blue banner explains that the Default value specifies what value the field has when first displayed. Below this, there is a checkbox for 'Use dynamic default' which is checked. The 'Dynamic default value' is set to 'Get Next Padded Number'.

- Setting Up Number Maintenance for the Daily Expenses Table

- Navigate to All in the application navigator.
- In the filter search bar, type Number Maintenance and select it from the list.
- Click on the New button to create a new number maintenance record.
- Provide the following details:
 - Table: Daily Expenses
 - Prefix: DFE
- Click on Submit to save the configuration.

The screenshot shows the 'Number Maintenance' form in ServiceNow. The form is titled 'Number Maintenance' and has a 'View: Advanced' dropdown. A blue informational banner at the top explains that dictionary entries manage how ServiceNow stores data in tables and fields (columns). It instructs users to select a Table and the field Type of the new column, enter a column label, and the column name. It also mentions setting a Max length for text String type fields, making the field Mandatory to save a record, and making the field a Display Value for reference fields.

The form fields are as follows:

- * Table:
- Prefix:
- * Number:
- Application:
- Number of digits:

Below the form fields are 'Update' and 'Delete' buttons. At the bottom, there is a 'Related Links' section with a link to 'Show Counter'.

- Configure the Form:

- Navigate to All in the application navigator.
- In the filter search bar, type Daily Expenses and select it from the results to open the module.
- Click on the New button to open a new record form.
- In the form header, right-click and select Configure > Form Design from the dropdown menu.
- In the Form Designer, use drag-and-drop functionality to customize the layout of the form fields as needed. Fields can be rearranged, moved between sections, or grouped based on specific requirements.
- Once the desired layout has been achieved, click Save or Close to apply the changes.
 - Before Form design:

servicenow All Favorites History Workspaces : Daily Expenses - Create DFE0001001

Daily Expenses
New record

Number DFE0001001 Date

Comments

Expense Family Member Name

Submit

Form design:

Daily Expenses [u_daily_expenses] 2 Column

Number Family Member Name

Date Expense

Comments

Submit

After form design:

servicenow All Favorites History Workspaces : Daily Expenses - Create DFE0001002

Daily Expenses
New record

Number DFE0001002 * Family Member Name

* Date Expense

Comments

Submit

Milestone 5: Creation of Relationship:

- Creation of Relationship between Family Expenses and Daily Expenses tables
 - Navigate to All in the application navigator.
 - In the filter search bar, type Relationships and select it from the list.
 - Click on the New button to create a new relationship.
 - Enter the following details:
 - Name: Daily Expenses

- Applies to Table: Family Expenses
 - Queries to Table: Daily Expenses
- Click Save to store the relationship.

Milestone 6: Configuring Related List on Family Expenses:

- Navigate to All in the main menu.
- In the filter/search bar, type Family Expenses, then select and open the Family Expenses module.
- Click on the New button to create a new Family Expenses record.
- In the header section of the screen, right-click and choose Configure from the context menu.
- From the configuration options, select Related Lists.
- In the Related Lists configuration screen, locate Daily Expenses in the available list.
- Move Daily Expenses to the Selected area to include it in the related lists for Family Expenses.
- Click Save or Done (depending on the interface) to apply the changes.

Milestone 7: Creation of Business Rules:

- From the main navigation, go to All.
- In the search or filter bar, type Business Rules.
- Under the System Definition section, click on Business Rules.
- Click on the New button to create a new business rule.
- Fill in the Business Rule Details:
 - Name: Family Expenses BR
 - Table: Select Daily Expenses from the dropdown list.
- Check the box labeled Advanced to enable additional configuration options.
- Configure When to Run: Under the When to Run section, make sure the following options are checked:
 - Insert
 - Update

servicenow All Favorites History Workspaces Business Rule - New Record

Business Rule New record

A 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. [More Info](#)

Name: Family Expenses BR Application: Global

Table: Daily Expenses [u_daily_expenses] Active: ☒ Advanced: ☒

When to run: When before Order: 100

Filter Conditions: Add Filter Condition Add "OR" Clause

Role conditions:

- In Advance (write the code): Write the below code >>

servicenow All Favorites History Workspaces Business Rule - New Record

Business Rule New record

Table: Daily Expenses [u_daily_expenses] Active: ☒ Advanced: ☒

When to run: When before Order: 100

Filter Conditions: Add Filter Condition Add "OR" Clause

Role conditions:

Condition:

Script: ☒ Turn on ECMAScript 2021 (ES12) mode

```
(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 })(current, previous);
```

Milestone 8: Configure the Relationship:

- From the main navigation, go to All.
- In the search bar, type Relationships, then open the Relationships module.
- Locate and open the Daily Expenses Relationship record.
- Set the Following Details:
 - For Applies to table, select Family Expenses from the dropdown list.
- After entering the script, click Update to save your changes.

The screenshot shows the ServiceNow interface for configuring a relationship. The top navigation bar includes 'servicenow', 'All', 'Favorites', 'History', 'Workspaces', and a search bar. The current page is titled 'Relationship - Daily Expenses'. Below the title bar, there are fields for 'Name' (set to 'Daily Expenses'), 'Application' (set to 'Global'), 'Applies to table' (set to 'Family Expenses [u_family_expenses]'), and 'Queries from table' (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 this, there is a section for the script with a toggle for 'Turn on ECMAScript 2021 (ES12) mode'. The script area contains the following code:

```
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 of the form, there are 'Update' and 'Delete' buttons, followed by a 'Related Links' section with a link to 'Run Point Scan'.