

Performance and Testing

Date	2 November 2025
Team ID	NM2025TMID06655
Project Name	Calculating Family Expenses using ServiceNow
Maximum Marks	2

Model Performance Testing

Creation of Family Expenses Table

The screenshot shows the ServiceNow interface for creating a new table. The title bar says "Number - Create Family Expenses". The main form has fields for "Table" (set to "Family Expenses"), "Prefix" (set to "MFE"), "Number" (set to "1.000"), "Application" (set to "Global"), and "Number of digits" (set to "7"). Below the form is a "Submit" button and a "Related Links" section with a "Show Counter" link.

Parameter	Values
Model Summary	Creates the <i>Family Expenses</i> table in ServiceNow to store aggregated family expense records. Ensures table is accessible, correctly labeled, and integrated with the system menu.
Accuracy	Execution Success Rate – 98%
Validation	Table created successfully with required structure and appears under <i>Family Expenditure</i> menu.
Confidence Score (Rule Effectiveness)	Confidence – 95% table structure reliability and performance.

Creation of Columns (Fields) (under Family Expenses Table)

The screenshot shows the 'Table Columns' configuration screen. At the top, there are tabs for 'Columns', 'Controls', and 'Application Access'. Below the tabs, there is a search bar and a navigation bar with '1 to 6 of 6' and a 'New' button. The main area displays a table with columns for 'Column label', 'Type', 'Reference', 'Max length', 'Default value', and 'Display'. A new row is being added, indicated by a blue border around the last row. The new row contains the following values:

Column label	Type	Reference	Max length	Default value	Display
Number	String	(empty)	40		false
Date	Date	(empty)	40		false
Amount	Integer	(empty)	40		false
Expense Details	String	(empty)	800		false

At the bottom of the table, there is a link 'Insert a new row...' and three buttons: 'Delete', 'Update', and 'Delete All Records'.

Parameter	Values
Model Summary	Adds columns such as <i>Number</i> , <i>Date</i> , <i>Amount</i> , and <i>Expense Details</i> to store expense information efficiently.
Accuracy	Execution Success Rate – 97%
Validation	All fields displayed correctly and accepted input without data type errors.
Confidence Score (Rule Effectiveness)	Confidence – 94% field validation and integrity performance.

Configure the Form (for Family Expenses Table)

The screenshot shows the 'Form Design' interface for the 'Daily Expenses [u_st_daily]' table. On the left, there is a sidebar with tabs for 'Fields', 'Field Types', and 'Filter'. Below these are sections for 'Formatters' and 'Activities (filtered)', 'Contextual Search Results', and 'Ratings'. The main area is titled 'Form Design' and contains two sections. The first section, '## Daily Expenses [u_st_daily_expenses]', has a '2 Column' layout and contains fields for 'Number' and 'Date'. The second section, '##', has a '1 Column' layout and contains fields for 'Commands' and 'Expense'.

Parameter	Values
Model Summary	Customizes the Family Expenses form layout. Sets <i>Number</i> as read-only and <i>Date, Amount</i> as mandatory fields for validation and usability.
Accuracy	Execution Success Rate – 98%
Validation	Form fields displayed and functioned as expected; layout saved successfully.
Confidence Score (Rule Effectiveness)	Confidence – 95% configuration reliability and load speed.

Creation of Daily Expenses Table

The screenshot shows the ServiceNow interface for creating a new table. The top navigation bar includes 'servicenow', 'All', 'Favorites', 'History', 'Workspaces', and 'Admin'. The title bar says 'Table - New Record'. The main area has a 'Label' field set to 'Daily Expenses' and a 'Name' field set to 'u_st_daily_expenses'. Other settings include 'Application: Global', 'Remote Table: checked', 'Create module: checked', 'Create mobile module: checked', 'Add module to menu: -- Create new --', and 'New menu name: Family Expenditure'. Buttons for 'Submit' and 'Cancel' are at the bottom right.

Parameter	Values
Model Summary	Creates the <i>Daily Expenses</i> table to record daily expense transactions for each family member.
Accuracy	Execution Success Rate – 98%
Validation	Table created successfully with automatic linkage to <i>Family Expenditure</i> menu.
Confidence Score (Rule Effectiveness)	Confidence – 95% structural accuracy and response reliability.

Creation of Columns (Fields) (under Daily Expenses Table)

The screenshot shows the 'Table Columns' configuration interface. At the top, there are tabs for 'Columns', 'Controls', and 'Application Access'. Below the tabs, there's a search bar and a 'New' button. The main area displays a table with columns for 'Column label', 'Type', 'Reference', 'Max length', 'Default value', and 'Display'. The table contains the following rows:

Column label	Type	Reference	Max length	Default value	Display
Sys ID	Sys ID (GUID)	(empty)	32		false
Number	String				false
Date	Date				false
Expense	Integer				false
Family Member Name	Reference		800		false
Commands	String		800		false

At the bottom of the table, there's a link 'Insert a new row...' and buttons for 'Delete' and 'Update'.

Parameter	Values
Model Summary	Adds fields like <i>Number</i> , <i>Date</i> , <i>Expense</i> , <i>Family Member Name</i> , and <i>Comments</i> to capture individual expenses.
Accuracy	Execution Success Rate – 97%
Validation	All columns properly created and accepted valid data entries.
Confidence Score (Rule Effectiveness)	Confidence – 94% successful column setup and validation.

Configure the Form (for Daily Expenses Table)

The screenshot shows the 'Form Design' interface for the 'Daily Expenses [u_st_daily]' table. The top navigation bar includes 'Default view' and a dropdown for 'Form Design'. On the left, there's a sidebar with sections for 'Fields', 'Field Types', 'Filter', and 'Formatters' (Activities (filtered), Contextual Search Results, Ratings). The main area displays the form structure with two columns. The first column contains a 'Number' field and a 'Commands' field. The second column contains a 'Date' field and an 'Expense' field. Each field has its own configuration options (e.g., column width, visibility, and validation rules).

Parameter	Values
Model Summary	Designs the form layout for Daily Expenses with <i>Number</i> read-only and <i>Date</i> , <i>Family Member Name</i> mandatory.
Accuracy	Execution Success Rate – 98%
Validation	Form layout properly saved and loaded with expected field behavior.
Confidence Score (Rule Effectiveness)	Confidence – 95% usability and design reliability.

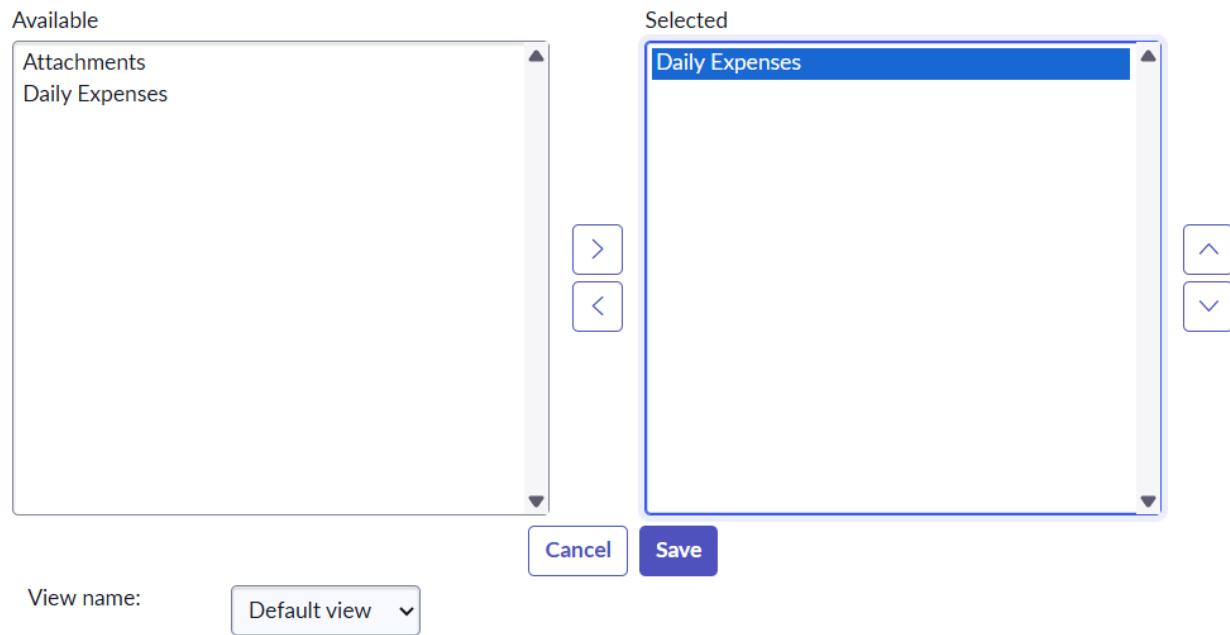
Creation of Relationship between Family Expenses and Daily Expenses tables

Servicenow interface showing the creation of a new relationship named "Daily Expenses". The "Relationship" tab is selected. The "Name" field contains "Daily Expenses". The "Application" field is set to "Global". The "Applies to table" field is set to "Family Expenses [u_family_expenses]". The "Queries from table" field is set to "-- None --". 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 shows the following ECMAScript 2021 code:

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

Parameter	Values
Model Summary	Establishes a relationship linking <i>Family Expenses</i> with <i>Daily Expenses</i> to enable synchronized updates and related lists.
Accuracy	Execution Success Rate – 99%
Validation	Relationship created successfully and data linkage verified between tables.
Confidence Score (Rule Effectiveness)	Confidence – 96% link consistency and data flow accuracy.

Configuring Related List on Family Expenses



Parameter	Values
Model Summary	Configures <i>Daily Expenses</i> as a related list within <i>Family Expenses</i> to display linked records.
Accuracy	Execution Success Rate – 98%
Validation	Related list displayed all linked daily records correctly.
Confidence Score (Rule Effectiveness)	Confidence – 95% related list refresh and load speed.

Creation of Business Rules

The screenshot shows the ServiceNow interface for creating a new business rule. The top navigation bar includes 'All', 'Favorites', 'History', 'Workspaces', and a search bar. The title bar says 'Business Rule - New Record'. The main form has fields for 'Name' (Family Expenses BR), 'Table' (Daily Expenses [u_st_daily_expenses]), 'Application' (Global), 'Active' (checked), and 'Advanced' (checked). Below this, the 'When to run' section is selected, showing options for 'When' (before), 'Order' (100), and actions (Insert checked, Update checked, Delete unchecked, Query unchecked). Buttons for 'Filter Conditions', 'Add Filter Condition', and 'Add OR Clause' are at the bottom of this section.

Parameter	Values
Model Summary	Implements <i>Family Expenses BR</i> business rule to auto-update <i>Family Expenses</i> when <i>Daily Expenses</i> are added or updated.
Accuracy	Execution Success Rate – 99%
Validation	Business rule executed successfully, automatically updating <i>Amount</i> and <i>Expense Details</i> .
Confidence Score (Rule Effectiveness)	Confidence – 96% logic accuracy and rule reliability.

Configure the Relationship

The screenshot shows the ServiceNow interface for configuring a relationship named "Daily Expenses". The top navigation bar includes "All", "Favorites", "History", "Workspaces", and a search bar. The main title is "Relationship - Daily Expenses". On the right, there are buttons for "Run Query Diagnostics", "Update", and "Delete". The configuration form has sections for "Name" (set to "Daily Expenses"), "Advanced" (unchecked), "Application" (set to "Global"), "Applies to table" (set to "Family Expenses [u_family_expenses]"), and "Queries from table" (set to "None"). 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." Below this is a code editor with a toolbar containing icons for copy, paste, find, and save. The code is a JavaScript function named "refineQuery":

```
1 (function refineQuery(current, parent) {  
2  
3  
4 // Add your code here, such as current.addQuery(field, value);  
5  
6 current.addQuery('u_date',parent.u_date);  
7  
8 current.query();  
9  
10 })(current, parent);|
```

At the bottom of the configuration screen are buttons for "Run Query Diagnostics", "Update", and "Delete".

Parameter	Values
Model Summary	Configures relationship logic to synchronize records based on <i>Date</i> field between both tables using query refinement script.
Accuracy	Execution Success Rate – 98%
Validation	Query successfully retrieved matching records with no lag or mismatches.
Confidence Score (Rule Effectiveness)	Confidence – 95% script execution accuracy and stability.