P06: Configuring a 3D Matrix Setup with Parent-Child Relationship in Dataverse

Monday, February 24, 2025 7:41 AM

Context Table:

Item	Detail
Tag	Vertical Development
Contributors	Vincent Kimiti
Version Control	3D Matrix Setup
Application Use	Structuring and managing related data across multiple tables in Dataverse
Reference Usage	Parent-Child Relationship in Dataverse for Subgrid Setup

Story Behind the Pattern:

Item	Detail
The Problem	Efficiently managing data within a parent-child relationship and enabling easy data entry and updates via subgrids.
The Solution	Set up a 3D matrix using a child table, establish the relationship with the parent, create necessary forms, and configure metadata for efficient record management.
Dependencies	N/A

The Pattern:

1. **Create Extra (3D) Table (Child Table)**:
   - Go to the **Dataverse(Powerpages)** portal and create a new table (child table) that will be used to store the related records.
   - This table should include the necessary columns to hold the data, including fields for the relationship to the parent table.
2. **Create Two Forms in the Child Table (CREATE & UPDATE)**:
   - **CREATE Form**: This form should allow users to create new records in the child table. Add necessary fields to capture the details of the child record.
   - **UPDATE Form**: This form should be used to update existing records in the child table. It should include all fields for editing child record details.
3. **Create Columns for the Subgrid in the Child Table**:
   - Add columns in the child table that will be included in the **subgrid** in the parent table form.
   - Ensure that these columns are relevant and contain the correct responding columns/questions related to the parent-child relationship.
4. **Create Relationship from the Parent Table**:
   - Set up a **one-to-many relationship** from the parent table to the child table.
   - This relationship will allow the child table's records to be linked and displayed in the subgrid on the parent table form.
5. **Add Sub-grid Component in the Parent Table Form**:
   - Open the **form** of the parent table and add the **sub-grid component**.
   - Link the subgrid to the child table using the **one-to-many relationship** created earlier.
   - Choose the columns to display in the subgrid, ensuring they match the ones created in the child table (e.g., fields that respond to the parent's data).
6. **Create a View for the Subgrid**:
   - Create a **view** for the child table records that will be displayed in the subgrid.

- Import this view into the parent table's subgrid component, ensuring the relevant columns are included and displayed in the correct order.

7. **Create Two Basic Forms (PPM) for Subgrid Data Management**:
   - **Basic CREATE Form**: This form should be used to add details to the subgrid by creating new child records linked to the parent.
   - **Basic UPDATE Form**: This form should allow users to update the child records directly within the subgrid.
   - These forms should support actions such as adding new records and updating existing ones.
8. **Configure Subgrid Metadata with 3 Actions (Create, Edit, Delete)**:
   - Create metadata for the subgrid that includes the following three actions:
     1. **Create**: Allows users to add new child records from within the subgrid.
     2. **Edit**: Enables users to update existing child records within the subgrid.
     3. **Delete**: Permits users to delete child records from the subgrid.
   - Ensure that these actions are properly linked to the corresponding forms (CREATE and UPDATE).
9. **Add Web Roles (Admin, Authenticated Users, etc.)**:
   - Define **web roles** for users who will interact with the subgrid.
   - Assign appropriate permissions (e.g., Admin, Authenticated Users) to enable different levels of access to the subgrid data.
10. **Table Permissions (Read, Write, Append)**:
    - Configure **table permissions** to control user access to the child table records.
    - Ensure that the appropriate roles (Admin, Users, etc.) have **read**, **write**, and **append** permissions on both the child and parent tables, allowing users to interact with the data as needed.