

Building “non-normal” User Interfaces While Keeping the Data Normal

Pete Poppe (login MajP on Access Forums)

2 July 2025

Background and Access

- Presented to Access Europe last August on using Tree Views, and you can reference that presentation for additional background
 - [Access Europe Treeviews 2024](#)
- Career Marine involved in defense acquisition / combat development then closely involved in similar work as a defense support contractor
 - Became Access power user / developer out of necessity to support my work and the customers I supported as defense contractor
 - System test results / test incidents
 - Survey analysis
 - Engineering requirements
 - Early 2000s little on-line access, remote test sites, small but repeatable applications
- Went to forums to get answers, and after a while could provide some
 - Active on Access World Forums because I enjoy the challenge of solving problems and answering questions
- Hobbyist, not something I do professionally

Problem Statement

- Certain access database requirements makes it seem very difficult to build a User Interface using native controls
 - Or at least it is hard to visualize how to build an interface
- Access users / developers often end up building non-normal data structures just to support a perceived required user interface.
 - Often referred to as “building Excel in Access”
 - Not limited to novice users but anyone can get stuck when you have more complicated structures
- Often the solution is to think a little “out of the box”

Common Problem Areas

- Spread Sheet (User desire to enter data like a spreadsheet)
 - Any Problems that intuitively we see as a grid (rows and columns)
 - If only Access had a dynamic Grid Control (Gridview)
- Scheduling with time slots (Outlook like appointment displays)
- Surveys (respondents, questions, answers)
 - An ultimate challenge in proper normalization and flexible display
 - Worth a separate / future Access Europe presentation
- Checklists (click from a set of selections)
- Calendar displays (month, year)
- Hierarchical or self referencing data

Objectives

- Present various tips, tricks, and work arounds I have used and seen other use
 - Not any “earth shattering code” but other ways to think about the interface
 - Presentation is more about a creative thinking than complex design
 - Many of these are techniques I have seen others do / not all original thought
 - Not a deep dive into any solution
- Caution / Caveat: Weigh these techniques when using
 - All presented methods have pros and cons. The native forms, subforms, and controls are well proven

Hopefully provide some tools to show users they do not have to force their data into non-normal structures just to support the perceived GUI

Common Techniques and Tools

- Crosstab for display with pop up to add/edit
- Preload records (insert query)
 - Example, Workflows
- ADO in-memory recordsets
- “Pseudo Grid” Control
- Temp tables
- Unbound Forms
- Graphical Displays “Pseudo Geo Reference”
- Tree views and “fake” tree views

Most techniques provide a user interface for Display and Navigation with a Pop-Up to add/edit/delete that is a normal bound form

Agenda

- Does not include all the techniques listed but does include:
 - Scheduling Problem
 - Unbound Forms
 - Checklists with Multiple Variants
 - Crosstab Grid
 - Cascading Combo Dilemma
 - Tree views and Fake Tree views
 - “Pseudo Geo Reference”
- Presentation a little long so hold all questions to end and put in chat
- If time permits, a period for open discussion at the end of what you have seen or done

The Scheduling Problem

- Specific Time Slots that user can schedule an appointment
 - Several Buildings
 - Several Conference Rooms per building
 - ½ hour time slots starting at 10am to 7pm
- User's original solution
 - A Table for each Room (26 tables)
 - A Field for each time slot (18 time slots)
 - A record is a day
- Solution
 - "Pseudo Grid" control

Store Data Normalized

tblRooms	refTblTimePeriod	tblAppointment
Field Name	Data Type	
autoReservationId	AutoNumber	
intPersonID	Number	
intRoomID	Number	
intApptID	Number	
dtmStartperiod	Date/Time	
dtmEndPeriod	Date/Time	
strReservationNotes	Long Text	
strReservationTitle	Short Text	

Clear and Fill

Fill Table to Present Data for that Day

autoRefTimePeriod	AutoNumber
strPeriodLabels	Short Text
dtmRefTimePeriodStart	Date/Time
dtmRefTimePeriodEnd	Date/Time
strRoom1	Long Text
strRoom2	Long Text
strRoom3	Long Text
strRoom4	Long Text
strRoom5	Long Text
strRoom6	Long Text
strRoom7	Long Text
strRoom8	Long Text
strRoom9	Long Text
strRoom10	Long Text
strRoom11	Long Text
strRoom12	Long Text
strRoom13	Long Text
strRoom14	Long Text
strRoom15	Long Text
strRoom16	Long Text
strRoom17	Long Text
strRoom18	Long Text
strRoom19	Long Text
strRoom20	Long Text

- Row for each time Period
- Column for each room to display
- Grid it out by adding all the labels and periods
- Select a day from the form
 - Clear old values for a last selected day
 - Populate with current selected day

“Pseudo Grid” is Continuous Bound Form built on a “Gridded” table for display and Navigation.
Displaying the data from a Normal Table

Actual Data

intPersonID	intRoomID	intApptID	dtmStartperiod	dtmEndPeriod	strReservationNotes	strReservationTitle
3	7	1	12	6/17/2025 10:00:00 AM	6/17/2025 10:29:00 AM This is a note	Some Event
4	7	2	22	6/18/2025 10:30:00 AM	6/18/2025 10:59:00 AM	Booking Two
5	14	1	7	6/17/2025 10:30:00 AM	6/17/2025 10:59:00 AM	Medical Appoint
6	2	2	14	6/17/2025 11:00:00 AM	6/17/2025 11:29:00 AM	Group Alcohol

Raw

dtmStartperiod	dtmEndPeriod	Details	intRoomID	TimeStart	TimeEnd	DateStart	strAlias
6/17/2025 10:00:00 AM	6/17/2025 10:29:00 AM	Sandy Holder - Keywork: Alcohol	1	10:00	10:29	6/17/2025	strRoom1
6/17/2025 10:30:00 AM	6/17/2025 10:59:00 AM	Julie Meade - Medical: Drug	1	10:30	10:59	6/17/2025	strRoom1
6/17/2025 11:00:00 AM	6/17/2025 11:29:00 AM	Mike Newman - SDC Group: Alcohol	2	11:00	11:29	6/17/2025	strRoom2

Query

autoRefTim	strPeriodLal	dtmRefTimePeri	dtmRefTimePerioc	strRoom1	Room2	strRoom3	strRoom4	strRoom5	strRoom
3	10:00 - 10:30	10:00:00 AM	10:29:00 AM	Sandy Holder - Keywork: Alcohol	Available	Available	Available	Available	Available
4	10:30 - 11:00	10:30:00 AM	10:59:00 AM	Julie Meade - Medical: Drug	Available	Available	Available	Available	Available
5	11:00 - 11:30	11:00:00 AM	11:29:00 AM	Mike Newman - SDC Group: Alcohol	Available	Available	Available	Available	Available
6	11:30 - 12:00	11:30:00 AM	11:59:00 AM	Available	Available	Available	Available	Available	Available
7	12:00 - 12:30	12:00:00 PM	12:29:00 PM	Available	Available	Available	Available	Available	Available
8	12:30 - 13:00	12:30:00 PM	12:59:00 PM	Available	Available	Available	Available	Available	Available
9	13:00 - 13:30	1:00:00 PM	1:29:00 PM	Available	Available	Available	Available	Available	Available
10	13:30 - 14:00	1:30:00 PM	1:59:00 PM	Available	Available	Available	Available	Available	Available
11	14:00 - 14:30	2:00:00 PM	2:29:00 PM	Available	Available	Available	Available	Available	Available
12	14:30 - 15:00	2:30:00 PM	2:59:00 PM	Available	Available	Available	Available	Available	Available
13	15:00 - 15:30	3:00:00 PM	3:29:00 PM	Available	Available	Available	Available	Available	Available
14	15:30 - 16:00	3:30:00 PM	3:59:00 PM	Available	Available	Available	Available	Available	Available
15	16:00 - 16:30	4:00:00 PM	4:29:00 PM	Available	Available	Available	Available	Available	Available
16	16:30 - 17:00	4:30:00 PM	4:59:00 PM	Available	Available	Available	Available	Available	Available
17	17:00 - 17:30	5:00:00 PM	5:29:00 PM	Available	Available	Available	Available	Available	Available
18	17:30 - 18:00	5:30:00 PM	5:59:00 PM	Available	Available	Available	Available	Available	Available
19	18:00 - 18:30	6:00:00 PM	6:29:00 PM	Available	Available	Available	Available	Available	Available
20	18:30 - 19:00	6:30:00 PM	6:59:00 PM	Available	Available	Available	Available	Available	Available

Transformed

Tuesday: 6/17/2025

VP Clinical rooms 1 - 4		VP Group rooms 1 - 4		VP Interview rooms 1 - 4	
DR Ground floor rooms 1 - 5		DR First floor rooms 1 - 6		VP HRP rooms 1 - 3	
DR Ground Floor 1	DR Ground Floor 2	DR Ground Floor 3	DR Ground Floor 4	DR Ground Floor 5	
10:00 - 10:30	Sandy Holder - Keywork: Alcohol	Available	Available	Available	Available
10:30 - 11:00	Julie Meade - Medical: Drug	Available	Available	Available	Available
11:00 - 11:30	Available	Mike Newman - SDC Group: Alcohol	Available	Available	Available
11:30 - 12:00	Available	Available	Available	Available	Available
12:00 - 12:30	Available	Available	Available	Available	Available
12:30 - 13:00	Available	Available	Available	Available	Available
13:00 - 13:30	Available	Available	Available	Available	Available
13:30 - 14:00	Available	Available	Available	Available	Available
14:00 - 14:30	Available	Available	Available	Available	Available
14:30 - 15:00	Available	Available	Available	Available	Available
15:00 - 15:30	Available	Available	Available	Available	Available
15:30 - 16:00	Available	Available	Available	Available	Available
16:00 - 16:30	Available	Available	Available	Available	Available
16:30 - 17:00	Available	Available	Available	Available	Available
17:00 - 17:30	Available	Available	Available	Available	Available
17:30 - 18:00	Available	Available	Available	Available	Available
18:00 - 18:30	Available	Available	Available	Available	Available
18:30 - 19:00	Available	Available	Available	Available	Available

June 2025

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5
6	7	8	9	10	11	12

Todays Date: Tue, 06/17/2025

Selected Date: Tue, 06/17/2025

Today

Selected date 6/17/2025

Go to Room

Reports

Print Schedule

Export to Excel

- Single subform bound to grid table
- Up to 6 Rooms
- Bind the controls to the correct columns when you select from tab
- Click a cell and pop up a Normal form bound to appointment table

Demo Room Booking

Form Header

IRooms", "strAlias = ""

IRooms", "strAlias = ""

IRooms", "strAlias = ""

IRooms", "strAlias = ""

blRooms", "strAlias =

blRooms", "strAlias =

blRooms", "strAlias =

Detail

strRoom1

strRoom2

strRoom3

strRoom4

strRoom5

strRoom11

Form Footer

Set Correct Room Captions

Bind Correct Textboxes

Another Pseudo Grid

Display Sport Equipment in a Three Column format based on Filter

SottoCategorie Attrezzature Atletica

Scelta SottoCategorie: A.D.2.1 - Lancio del Disco_Dischi per Allena

Scelta Tipo Prodotto:

☐ CasellaControllo23

Apri Report



12380_DI



BounceBall



Dumbbell



Golf Ball



HRD-1.5_SI



HRD-1.6_SI



Kettlebell



Ping Pong Paddle



Plate



Field Name	Data Type
ID	AutoNumber
ImagePath1	Short Text
ImageName1	Short Text
ImagePath2	Short Text
ImageName2	Short Text
ImagePath3	Short Text
imageName3	Short Text

- Differs from the scheduling since rows not known must add and delete rows and this can cause bloat if not handled
 - Compact repair
 - Side database where your copy, use, then delete
 - In-Memory Recordset
- In this case delete all records and read the normalized data
- Fill the table row by row column by column adding a row as you go

Spreadsheet Interface

- Often that is what the user wants to see data in columnar display
- With the idea of moving Normal data into a Temp Table you can mimic functionality of a Spread Sheet
- For simplicity I have shown add/edit/delete from a click to a Pop-up
- With some creativity
 - Edit directly in cells and write back / delete from data table
 - Navigate between cells using arrow keys
 - Allow filter and Sort

Using a Table as the Grid vs Unbound Form

- When the Number of rows and columns is few then a pure unbound form maybe simpler than using the “Pseudo Grid” table
 - The scheduling problem was 18 rows (time slots) by 6 max rooms = 108 controls
- If the number of rows and columns could change then more flexibility with the pseudo-grid idea
 - Modify or add the time slots only requires changes in the table
 - Unbound form would require code changes and form changes
- Pure unbound form solutions for large grids can be done by leveraging code to show, hide, and move controls

Unbound Form

- Caveat – Using unbound forms when you cannot do something bound. Not using unbound forms for basic CRUD
- Unbound for when the number of controls is known and limited.
- The Calendar is probably the most built and used
 - Data is normal and uses pop up to enter data
 - However, the novice user can produce some crazy non normal solutions to match this display

The screenshot shows a 'Schedule Calendar' application window. The main window has a title bar with a close button and the text 'Schedule Calendar'. Below the title bar, there is a navigation area with a '<<' button, a 'Month:' label, and a dropdown menu showing 'June'. To the right of the month selector is a 'Print' button. The calendar grid displays days of the week (Sunday through Saturday) and dates. The date '3' under 'Tuesday' is highlighted. A pop-up window titled 'Input Box' is overlaid on the calendar, with a subtitle 'Tuesday June 3, 2025'. The input box contains the text 'Add a Meeting on 3 June' and has 'OK', 'Delete', and 'Cancel' buttons at the bottom. The bottom right of the main window displays 'June, 2025' in a large, dark red font.

Same idea just Extended

- Year view using multiple subforms and Real Text to color multiple events
- Normal data (start, end, absence type)
- Click to add / edit an absence/event

January							February							March							April							May							June						
Sa	Su	Mon	Tue	Wed	Thu	Fri	Sa	Su	Mon	Tue	Wed	Thu	Fri	Sa	Su	Mon	Tue	Wed	Thu	Fri	Sa	Su	Mon	Tue	Wed	Thu	Fri	Sa	Su	Mon	Tue	Wed	Thu	Fri							
			1	2	3		1	2	3	4	5	6	7				1	2	3	4						1	2				1	2	3	4	5	6					
4	5	6	7	8	9	10	8	9	10	11	12	13	14	8	9	10	11	12	13	14	5	6	7	8	9	10	11	3	4	5	6	7	8	9	7	8	9	10	11	12	13
11	12	13	14	15	16	17	15	16	17	18	19	20	21	15	16	17	18	19	20	21	12	13	14	15	16	17	18	10	11	12	13	14	15	16	14	15	16	17	18	19	20
18	19	20	21	22	23	24	22	23	24	25	26	27	28	22	23	24	25	26	27	28	19	20	21	22	23	24	25	17	18	19	20	21	22	23	24	25	26	27			
25	26	27	28	29	30	31								29	30	31					26	27	28	29	30			24	25	26	27	28	29	30	28	29	30				

July							August							September						
Sa	Su	Mon	Tue	Wed	Thu	Fri	Sa	Su	Mon	Tue	Wed	Thu	Fri	Sa	Su	Mon	Tue	Wed	Thu	Fri
			1	2	3	4							1			1	2	3		
5	6	7	8	9	10	11	2	3	4	5	6	7	8	6	7	8	9	10		
12	13	14	15	16	17	18	9	10	11	12	13	14	15	13	14	15	16	17		
19	20	21	22	23	24	25	16	17	18	19	20	21	22	20	21	22	23	24		
26	27	28	29	30	31		23	24	25	26	27	28	29	27	28	29	30			
							30	31												

Update Absence

Tuesday, May 27, 2014Cat, Bob

(EA) Excused Absence

Enter AbsenceTime: 1.00

Reason: Dental

Enter AbsenceTime:

Reason:

OK

Crosstab as Grid

- Timely example while developing this Presentation
 - [AccessWorld Forums 24 Jun](#)
- User has Insurance Quotes by Company with each Company as a column
 - Multiple experts showed how to normalize data
 - Customer Table
 - Company Table
 - Junction Table
 - Showed a Customer form and Quote Subform

I can maintain the simplicity of entry I want and still get the answers that I need.

I am not disputing the correctness of what was offered. You guys are right on. It's simply a command decision on my part that I'm making for this particular application. Even though not technically correct, I think it will provide me with the most benefit overall. AWF 24 Jun

Crosstab with Popup

Customer Quotes

CustomerName	Min Quote	Geico
John Smith	\$200.00	
Pam Brown	\$123.00	
Mark Black		
Jody Green		

CustomerID: 26
CustomerName: John Smith
Phone:

Quotes

Company Name	Quote
Geico	
Hagerty	
Liberty Mutual	
Progressive	
State Farm	

- Alternate solution was to Preload companies in Junction Table
 - Have 5 companies marked as default in the Companies table
 - Insert those 5 records on a new customer entry
 - Traditional sub form but already pre-loaded

The Checklist

- Users want to select from a list of things for a given record
 - Activities to complete
 - Yes / No accomplishments
 - Inspections
 - Orders, ingredients, components
- Users do not want to create a child record one by one
 - Pull down from a combo box in subform
 - Hard to see all possible choices and what has been selected
- User will often create a column for each choice creating a non-normal structure
 - Time consuming (lots of columns)
 - Not flexible (add, modify, or delete choices requires redesign of table and forms)

Checklist

- Almost all are Many to Many
 - A Selections Table
 - A People or Items data table
 - A Junction table Selection ID, Data Item ID

- A Junction table Selection ID, Data Item ID

Many Variations but all Similar

- 1. Table of selections to choose from
- 2. Normalized junction table with selected item
- 3. When form loads, clear existing selections from display, then populate display with “user” selected items
- 4. Select or deselect from display. Normally a click
- 5. Write to or delete from junction table

```
Private Sub Form_Current()
    ClearSelections
    LoadSelections
End Sub
```

Current Event Clear selections from Display
then Re-add them

```
Public Sub LoadSelections()
    'For Multi select
    Dim strSql As String
    Dim rs As DAO.Recordset
    Dim i As Integer
    strSql = "Select * from tblUserSelections where UserID_FK = " & Nz(Me.UserID, 0)
    Set rs = CurrentDb.OpenRecordset(strSql)
    Do While Not rs.EOF
        For i = 0 To Me.lstSelections.ListCount - 1
            If CLng(Me.lstSelections.Column(0, i)) = rs!SelectionID_FK Then
                Debug.Print rs!SelectionID_FK & " " & Me.lstSelections.Selected(i) = True
                Me.lstSelections.Selected(i) = True
            End If
        Next i
        rs.MoveNext
    Loop
    Me.Refresh
End Sub
```

Read the Junction table and load for
selected main record

```
Public Sub ClearSelections()
    Dim i As Integer
    For i = 0 To Me.lstSelections.ListCount - 1
        Me.lstSelections.Selected(i) = False
    Next i
End Sub
```

Regardless of User
Interface and controls
used, it is some version of
this.

```
Private Sub lstSelections_Click()
    If Me.lstSelections.Selected(Me.lstSelections.ListIndex) = True Then
        AddSelection Me.lstSelections.Column(0, Me.lstSelections.ListIndex)
    Else
        RemoveSelection Me.lstSelections.Column(0, Me.lstSelections.ListIndex)
    End If
    ClearSelections
    LoadSelections
End Sub
```

When you select an item determine
if you add or remove

```
Public Sub AddSelection(SelectionID As Long)
    Dim strSql As String
    strSql = "Insert into tblUserSelections (UserID_FK, SelectionID_FK) VALUES (" & UserID & ", " & SelectionID & ")"
    CurrentDb.Execute strSql
End Sub
```

Insert to add item from junction table

```
Public Sub RemoveSelection(SelectionID As Long)
    Dim strSql As String
    strSql = "Delete * from tblUserSelections WHERE UserID_FK = " & UserID & " AND SelectionID_FK = " & SelectionID
    CurrentDb.Execute strSql
End Sub
```

Delete to remove from Junction table

Demo

- Using a Subform as the UI
- Using a Multiselect Listbox
- Using an In-memory ADO Recordset bound to a Form
- Using a Form based on a CrossTab Query

UserID	UserName	Click to Add
1	John Smith	
2	Mike Brown	
3	Dave Black	
4	Jones	
*	(New)	

SelectionID	SelectionText	Selected	Category
1	A: Pre-engagement Activity One	☐	Pre-Engagement
2	B: Pre-engagement Activity Two	☒	Pre-Engagement
3	C: Pre-engagement Activity Three	☒	Pre-Engagement
4	D: Pre-engagement Activity Four	☐	
5	E: Mid-engagement Activity One	☒	
6	F: Mid-engagement Activity Two	☐	
7	G: Mid-engagement Activity Three	☒	
8	H: Mid-engagement Activity Four	☒	
9	I: Post-engagement Activity One	☐	
10	J: Post-engagement Activity Two	☐	
11	K: Post-engagement Activity Three	☐	
12	L: Post-engagement Activity Four	☐	

ID	UserID_FK	SelectionID_FK	Click to Add
108	4	4	
110	4	6	
111	3	2	
113	2	2	
114	2	3	
115	2	5	
116	2	7	
117	2	8	
127	1	10	
128	1	11	
129	1	12	

Demo
Checklists

Creating In Memory Recordset

- Add any Fields / Columns you want and bind to a form
- Same idea as a “Pseudo Grid” where you create a non-normalized display and populate with normal data
- Benefit of not requiring a table
 - No bloating from adds and deletes like a temp table
- Most of the time I really only need a few fields
 - PK field that relates to the PK of the normal data
 - Display Field to show something (Often concatenated data)
 - Boolean field to be able to check / uncheck

```

Public Sub CreateInMemoryRecordset(Domain As String, ValueField As String, Optional PK_Field As String = "None")
    'Either Storing the PK or the value field in linked table
    'Ensure reference to ADODB, Active Data objects
    Dim rsFill As DAO.Recordset
    Dim strSql As String

    With InMemoryRS
        .Fields.Append "Selected", adBoolean
        .Fields.Append "ValueField", adVarChar, 255, adFldMayB
        .CursorType = adOpenKeyset
        .CursorLocation = adUseClient
        .LockType = adLockPessimistic
        '.Open
    End With

    If Not PK_Field = "None" Then
        strSql = "SELECT " & PK_Field & ", " & ValueField & " from " & Domain
        InMemoryRS.Fields.Append "PK_Field", adBigInt ' currently only handle long PKs
    Else
        strSql = "SELECT " & ValueField & " from " & Domain
    End If
    InMemoryRS.Open
    ' MsgBox strSql
    Set rsFill = CurrentDb.OpenRecordset(strSql)
    Do While Not rsFill.EOF
        InMemoryRS.AddNew
        InMemoryRS.Fields("ValueField") = rsFill.Fields(ValueField)
        If Not PK_Field = "None" Then InMemoryRS.Fields("PK_Field") = rsFill.Fields(PK_Field)
        InMemoryRS.Update
        rsFill.MoveNext
    Loop

    Me.txtValue.ControlSource = "ValueField"
    Me.chkSelected.ControlSource = "Selected"
    Set Me.Recordset = InMemoryRS
    Me.Recordset.MoveFirst
End Sub

```

In-Memory RS is an ADODB.Recordset
declared at top of module

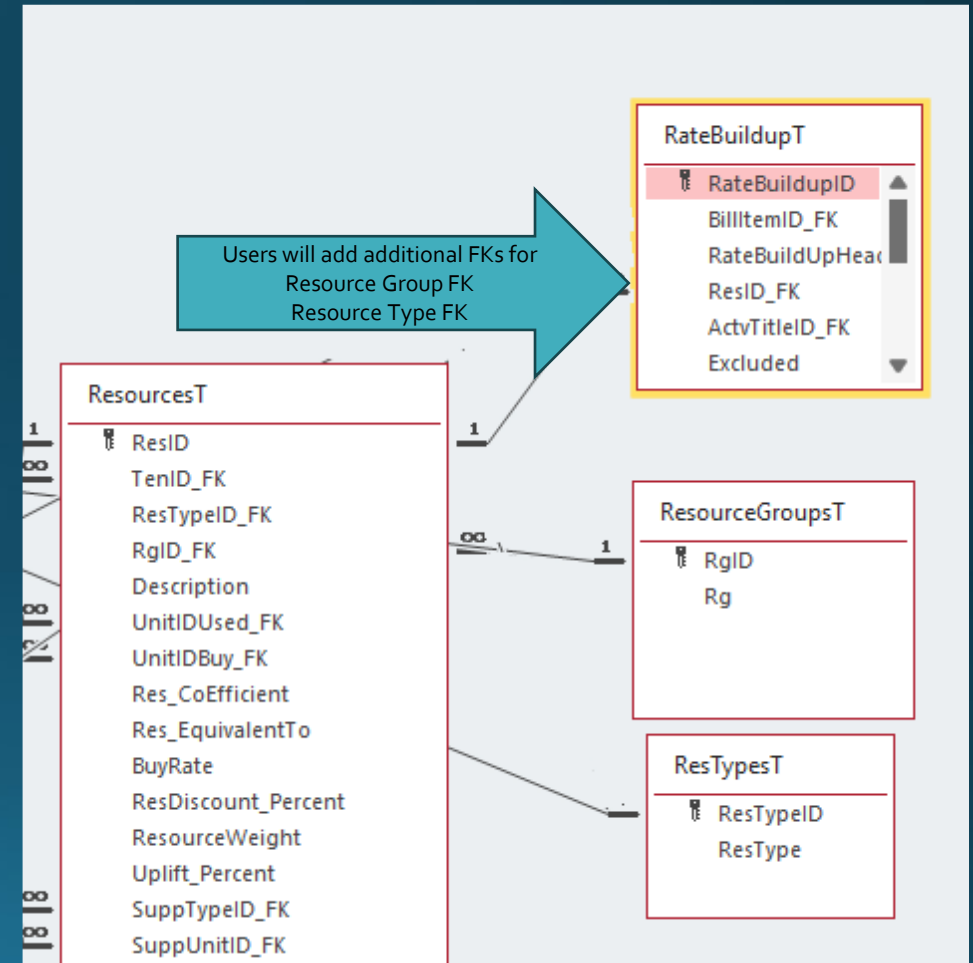
Use Append method to add fields.
Selection Field, Value (Display Field)
PK field for selection

Fill the In-Memory RS with your possible
selections

The screenshot shows a Microsoft Access form in design view. The form has a table with two columns: 'ValueField' and 'Selected'. The 'Form Header' section contains two rows: 'Sort A to Z' and 'Sort Z to A', each with a checkbox and a dropdown arrow. The 'Detail' section shows the 'ValueField' column with a checkbox. The 'Form Footer' section contains 'OK' and 'Cancel' buttons.

Cascading Combo Dilemma

- Problem Statement: User have a long list of choices and choices fall into Categories / Groups
- To support cascading comboboxes, especially in continuous form users store additional foreign keys in data table
- I see this problem with seasoned database builders as well as novices
 - (I see this as an issue, but others do not seem concerned)
- Problem because:
 - Adding unnecessary data fields to support the UI
 - Additional FKs can get out of synch
 - Circular reference (Data table stores a foreign key to Group / Type but also through the Resource relationship)



Solution – Change your Mindset

Do not Cascade. Use a Popup

- More robust and flexible than CC
- Allows correct normalization
- Doing cascading with Unbound combos in continuous form is hard

1.1 - Reduce lv's

Select Resources

Copy RB FROM BI

Copy RB TO BI

Bill Item	Ref	Constant	Output	Waste_Percent	Resource	Adjust Rate	Adjust Qty
Reduce lv's							
Reduce lv's							
*							

Search for Resource and Activities

Test Tender

(Double Click or Hit OK to select Current Resource)

Resource Ty	Resource Gr	Description	Unit	ResRt
Labour	In-House	Machine Driver	Hrs	£18.00
Material	Aggs & Muck	Non-Hazardous Off-Site	m3	£31.18
Material	Aggs & Muck	Sharp Sand (Bulk)	m3	£33.30
Material	Aggs & Muck	Reject Sand (Bulk)	m3	£25.20
Material	Aggs & Muck	Type 1 (Bulk)	m3	£59.54
Material	Fuely Bo	Diesel	Ltrs	£1.19
Plant	Excavators	13t Excavator	Hrs	£11.56

OK/Add

Close

Copy

Resource

Machine Driver

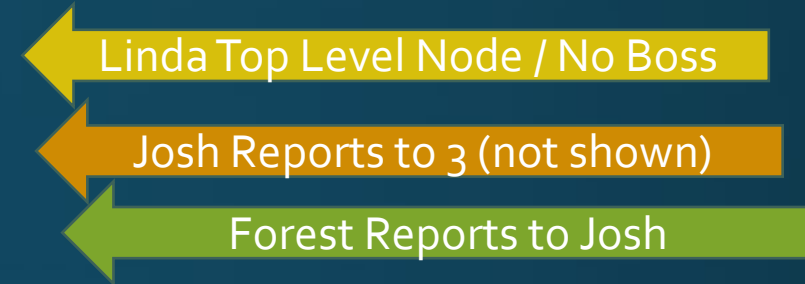
Hierarchical Data and Tree Views

- This Presentation just to remind you that this tool provides an intuitive means to see, process, and work with this type of data.
 - Keeps users from creating complex data structures to support the GUI
 - See Previous presentation: [Access Europe Treeviews 2024](#)
- Hierarchical data is hard to display, visualize, and work with using native forms and controls
 - But self-referencing tables are extremely easy to build
- Big Fan of Tree Views
 - IMO no better way to display and make the hierarchy understandable
 - IMO no easier way to navigate and work with data
- Common Examples
 - Family trees / genealogy
 - Bill of Materials / Assemblies and parts
 - Tasks-subTasks
 - Anything with numerous and indefinite levels, groups
- Bottom Line
 - If you have Hierarchical data, then a tree view allows an interface and keep data normal by using a self referencing table
 - You can fake a hierarchical display with a few tricks but requires the same complexity as loading a tree
 - Use in standard form
 - Use in a report
 - Use in Listbox

Challenge

Hierarchical Data often Self Referencing

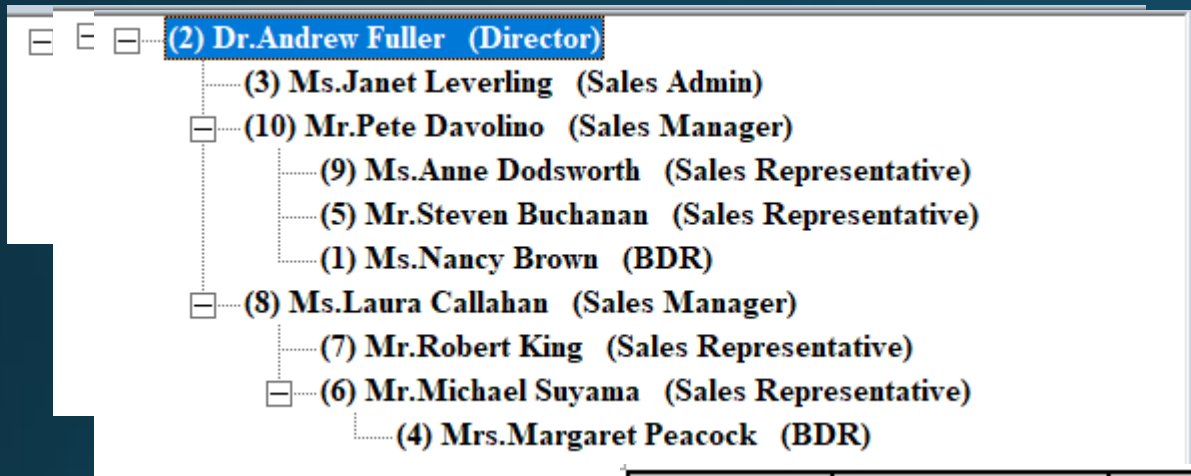
Person_ID	Full_Name	Boss_ID
0	Linda Powlowski	
1	Madilyn Osinski	
4	Joshuah Parisian	3
5	Forest Oberbrunner	4
9	Gwen Wunsch	8
10	Ona Christiansen II	11



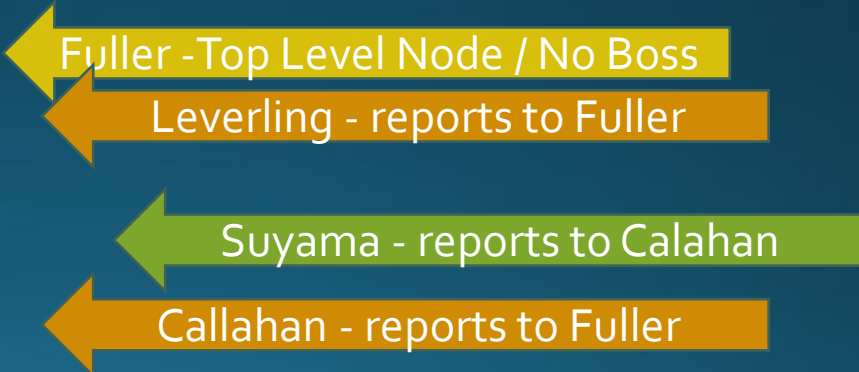
- Building the table / relationship is simple but users cannot figure out how to display / work with
- Users will then do the work around of guessing how many levels and make tables
 - Parent Table, Child Table, Child-Child Table, Child-Child-Child table.....
 - Data structure may not work (Employee A is 1st line supervisor for some and second for others)
 - Data structure requires you to know the levels and relations
 - Make complex forms with sub, sub-sub, sub-sub-sub...forms
 - Unmanageable after three levels
 - Not doable with unknown / unlimited levels

Intuitive Display

But almost Impossible to decipher when looking at the data level



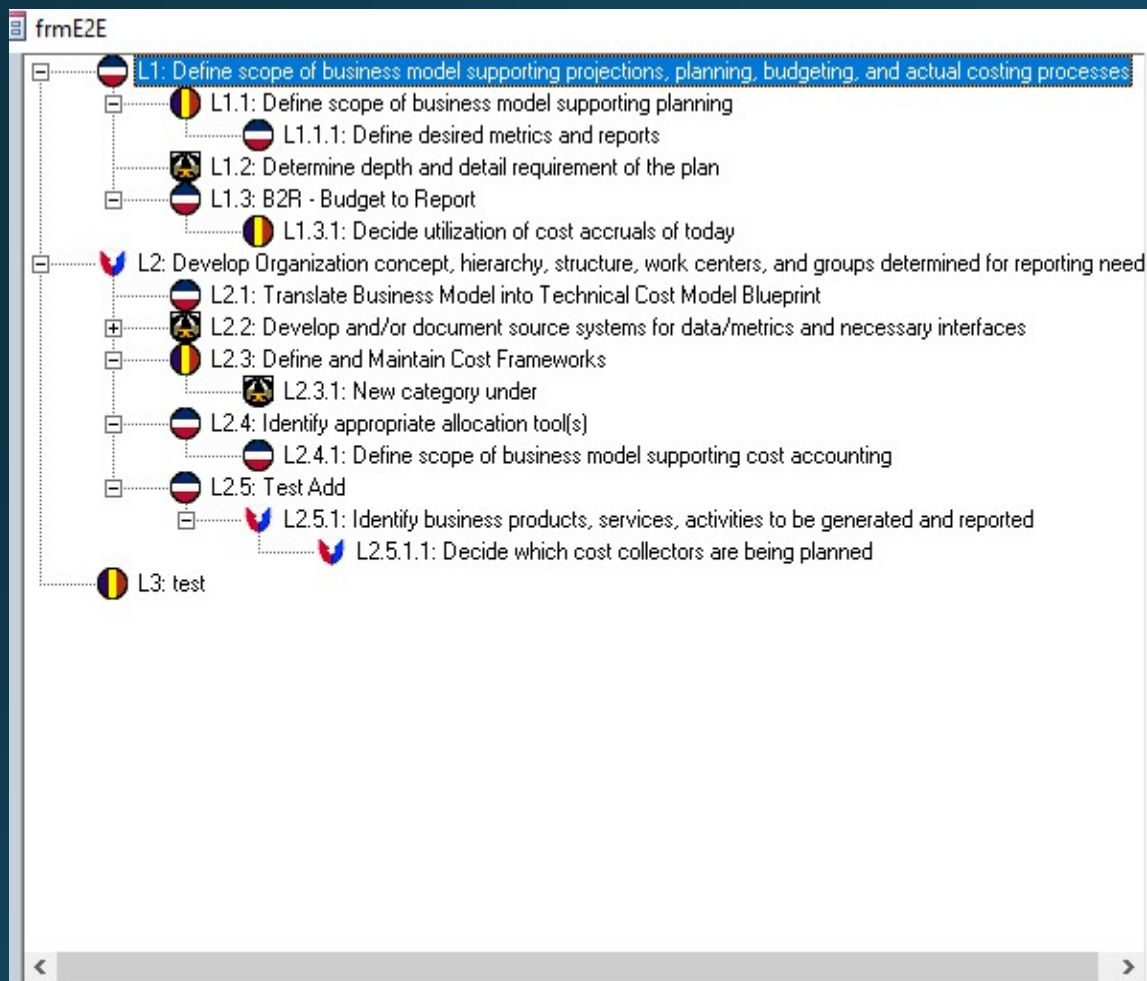
Employee ID	Last Name	Reports To
1	Brown	10
2	Fuller	
3	Leverling	2
4	Peacock	6
5	Buchanan	10
6	Suyama	8
7	King	8
8	Callahan	2
9	Dodsworth	10
10	Davolino	2



Think how you would display and work with this with native forms

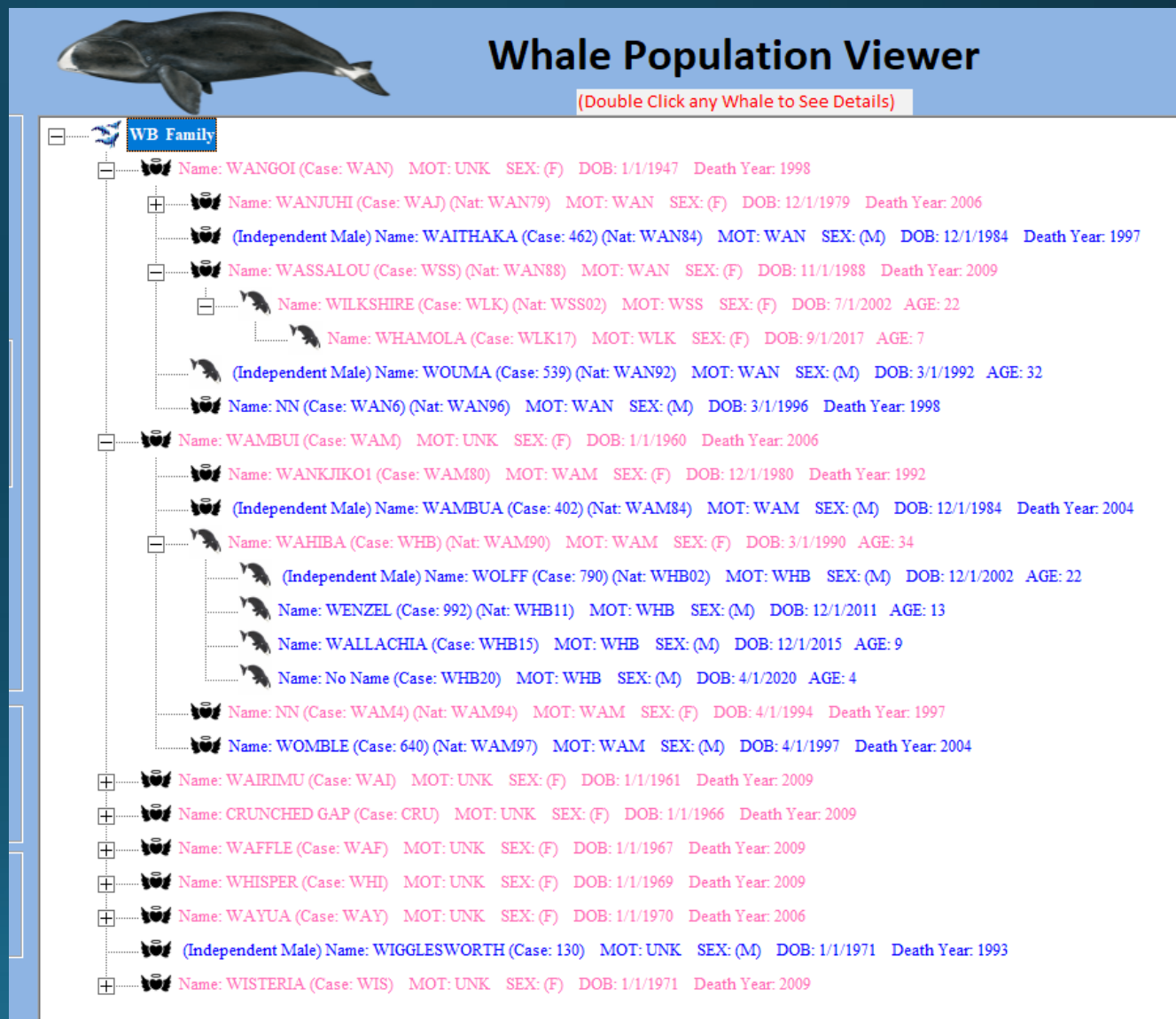
Tree View Examples

Operational Tasks-Sub Tasks with Icons



Features: Promote, Demote, Add at level, Delete

Whale Pod Genealogy icons and colors





Access Item Genie

LOCATIONS

Total Value
\$12,998.20

Find Item



TREEVIEW MENU

Add / Delete Nodes

---Locations---

Add Location (sub-location)

Add Location (Root)

---Items---

Add Item (Located)

Add Item (Not Located)

Add / Edit Bulk Items

---Delete---

Delete Node

Move Nodes

Move Current Node to Root

---Bulk Move---

Check Nodes to Move

Position Nodes

▲ Move Up

▼ Move Down

Expand / Collapse Tree

Expand Tree

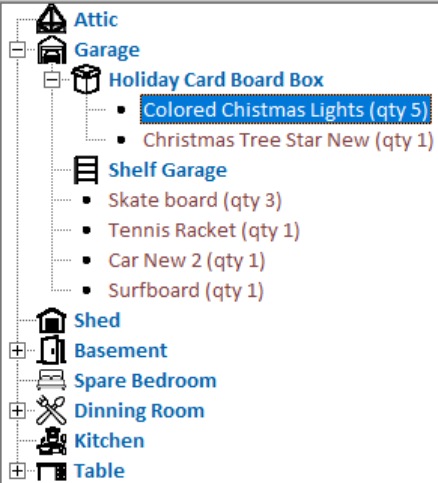
Collapse Tree

Expand Branch

Collapse Branch

Application Settings

Application Settings



Items in Holiday Card Board Box Item Details Summary Data

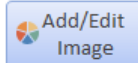


Items Details

Item ID 6

Item Name Colored Chistmas Lights

Item Description



(Double Click loaded image to Zoom)

Location Holiday Card Board Box

Category

Purchase Cost \$12.00

Purchase Date 2/19/2025

Current Value \$12.00

Quantity 5

Manufacturer Lights.com

Model Number ABC

Serial Number 12345

Item Notes

Delete Item

Demo Item Genie
Treeview

Demo if Time Permitting

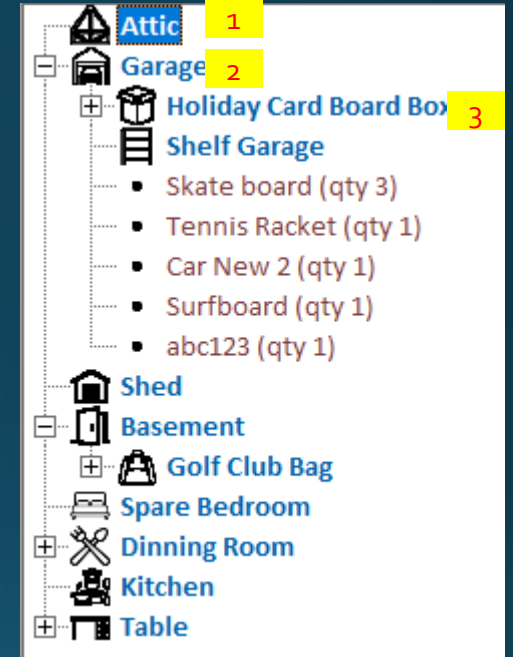
Fake Hierarchical Reports and Forms

Store a “TreeSort” and “Node Level”

Location_ID	LocationName	Parent_ID	TreeSort	NodeLevel
40	Attic		1	1
1	Garage			
3	Holiday Card Board Box	1	3	2
36	Shelf Garage	1	6	2
39	Shed		12	1
38	Basement		13	1
2	Golf Club Bag	38	14	2
37	Pouch	2	15	3
42	Spare Bedroom		18	1
53	Dinning Room		19	1
41	Office	53	20	2
44	Left Shelf	41	21	3
43	Right Shelf	41	23	3
45	Desk	41	25	3
46	Kitchen		27	1
54	Table		28	1

Top Level location

Inside the Garage



Tree Sort (from top down)

Node level (how it is indented)

Fake Tree examples in form and Report

10.1	---10.1 - Our contract ends upon pouring concrete with the exception that our works are
11	11 - Scope Related
11.1	---11.1 - Setting-Out
11.1.1	-----11.1.1 - Ltd to our works - inc'd
11.1.2	-----11.1.2 -
11.2	---11.2 - Drainage
11.2.1	-----11.2.1 - Based on PVCu u.n.o
11.2.2	-----11.2.2 - Attenuation Tanks - often over spec'd; allowed for 400N tanks
11.2.3	-----11.2.3 - Gullies
11.3	---11.3 - Groundworks
11.3.1	-----11.3.1 - Disposal - based on inert material off-site
11.3.2	-----11.3.2 - Water discharged into mains < 200m of our works. Assu
11.4	---11.4 - Disposal
11.4.1	-----11.4.1 - Based on inert
11.4.2	-----11.4.2 - Surface water inc'd; based on discharging into mains
11.4.3	-----11.4.3 - Surface water - assumed meets with dischargal criteria;
11.4.4	-----11.4.4 - Groundwater - exc'd
11.4.5	-----11.4.5 - Surface water - dischargal permits (if required) by client
11.5	---11.5 - Cranes
11.5.1	-----11.5.1 - Exc'd; by client
11.5.2	-----11.5.2 - Void infills - price subject to action < 1 year from our co
11.6	---11.6 - Reinforced Concrete
11.6.1	-----11.6.1 - Reinforcement

- Cannot be done with Sorting and Grouping due to unknown levels
 - Or extremely complicated
- May be impossible to mimic sort order

10- (H) Maintenance
10.1- (C) Our contract ends upon pouring concrete with the exception that our works are to standards acceptable by NSCS & relevant governing bodies
11- (H) Scope Related
11.1- (H) Setting-Out
11.1.1- (C) Ltd to our works - inc'd
11.1.2- (C)
11.2- (H) Drainage
11.2.1- (C) Based on PVCu u.n.o
11.2.2- (C) Attenuation Tanks - often over spec'd; allowed for 400N tanks
11.2.3- (H) Gullies
11.3- (H) Groundworks
11.3.1- (C) Disposal - based on inert material off-site
11.3.2- (C) Water discharged into mains < 200m of our works. Assumed water meets local authorities dischargal criteria. No allowance for treating to meet dischargal criteria
11.4- (H) Disposal
11.4.1- (C) Based on inert
£0.00

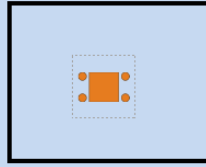
Graphical Interface / “Geo Locate”

- Challenge to locate Items on an image (map)
- Save image to table and redisplay
- Three attempts
 - Restaurant layout as part of Point-of-Sale system
 - Tasks to complete on a boat
 - Risks located in a factory

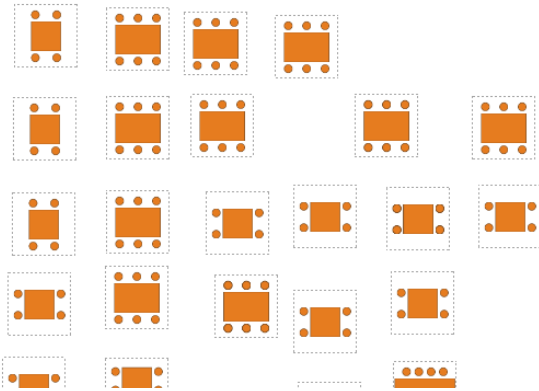
Select Floor Plan

Café (binnen)

Save Plan



Drag Above Image
 onto Floor
 (Double Click
 to Change Image Type)



job) onto deck plan and enter the nformation in the job pane

lete or Edit Job. Single Click to show details.)

FloorName

Profile

Description

Bad circuit

Department

Galley

Reported Date

11/20/2023

Reported By

Paul

1. Select Site Location

Участок 5 ПУШП

2. Select Site Map

Site Map Location X SubLocation 2

Unlocated Risks

Risk Filter

1A Умеренный

1B Умеренный

1C Умеренный

2A Умеренный

2B Умеренный

2C Умеренный

3A Умеренный

1D Значительный

1E Значительный

2D Значительный

3B Значительный

3C Значительный

4A Значительный

2E Высокий

3D Высокий

4B Высокий

4C Высокий

5A Высокий

5B Высокий

23 Критический

4D Критический

4E Критический

5C Критический

5D Критический

5E Критический

☐ Moderate

☐ Significant

☐ High

☐ Critical

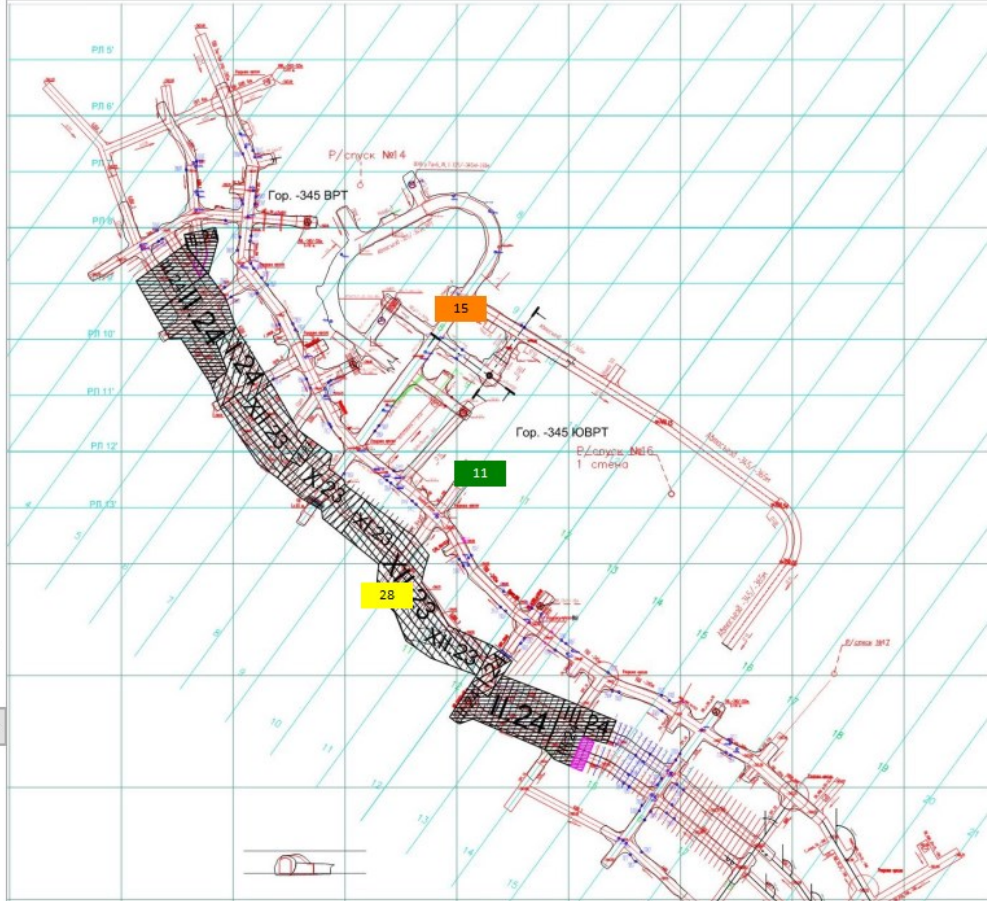
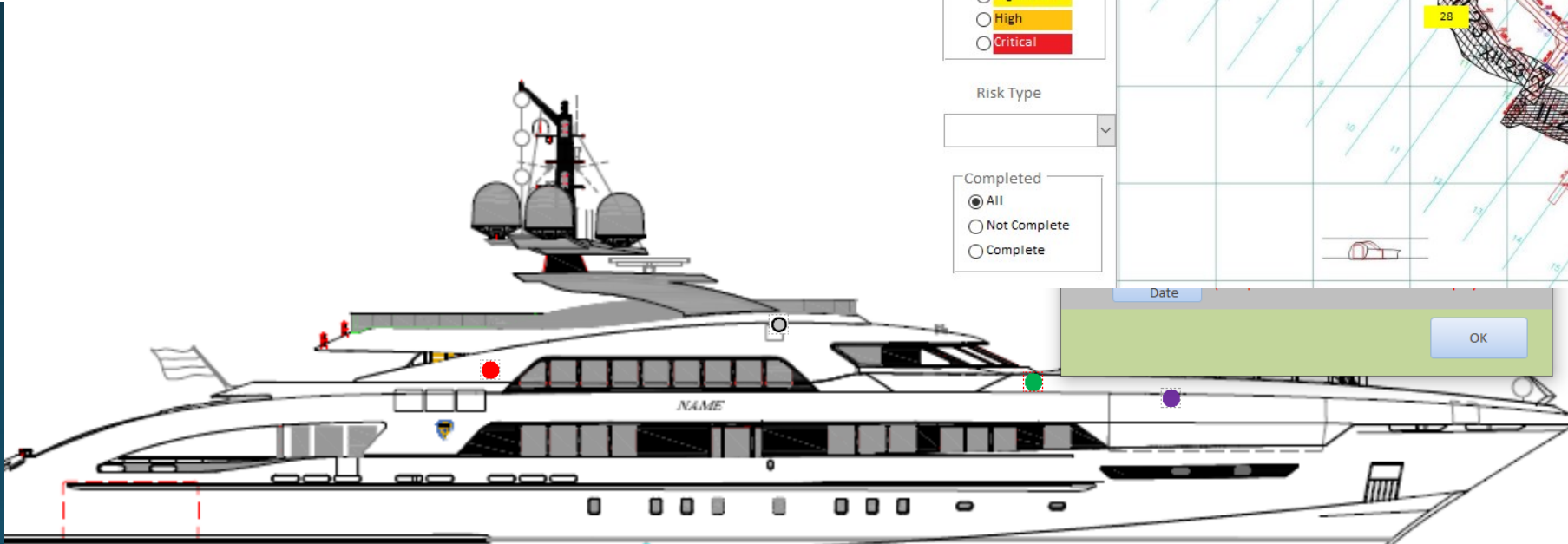
Risk Type

Completed

☒ All

☐ Not Complete

☐ Complete

Demo Risks

Mapping

- Not as complicated as it looks
- Table just stores
 - X_Pos 'Left
 - Y_Position 'Top
 - Associated image
 - (NOTE: no real geo reference, only related to that specific image at that scale)
- Class Module for "MoveableLabel"
 - Instantiate a variable when you click on a label handles the moving
- Trick
 - Create a lot of hidden labels and stack them up
 - Read the table pick one of the hidden labels and move it into place
 - After you "drag and drop" update the X, Y pos

Links on AWF

- Treeviews
 - [Item Genie](#)
 - [Hierrarchical Data, Self-referencing, Treeviews](#)
- Schedule, “Pseudo Grid”
 - [Schedule Grid](#)
- Calendars, Date Pickers
 - [Calendars and Date Pickers](#)
- “Geo Reference” (drag drop on image)
 - [Risks](#)
 - [Restaurant](#)
 - [Ship Tasks](#)
- Deep Dive into a “Non-Normal” Form
 - See Pat Hartman’s presentation [Access User Groups / Pacific October 2023](#)

Questions?

- Pseudo Grids
- Checklists
- Crosstabs
- ADO In-Memory
- Unbound Displays
- Graphical “geo locate”
- Treeviews and Fake Tree Views