# Project 1 – Training Attendance Summary

This Power BI project was developed to replicate a solution I had built for a previous employer, where I was responsible for tracking attendance across mandatory employee training sessions. This project demonstrates advanced-level DAX modeling, data shaping using Power Query, and hierarchical role mapping.  
The dataset includes TAO, BAO, and Delegate roles mapped with 1-UP managers. The solution includes two main DAX-based tables:

* Training Progress Tracking
* Final Attendance Summary

The core of this project revolves around the following advanced DAX logic:

## DAX Table: Training Progress Tracking

Key elements in the code:

* Uses UNION and SELECTCOLUMNS to combine trainee-role-manager mappings.
* Filters data using EARLIER to extract meeting attendance status.
* Uses CONCATENATEX for aggregating DIAMOND MAL Codes by trainee.
* Handles multiple roles (TAO, TAO Delegate, BAO) with respective managers.

Training Progress Tracking =

**-- Step 1: Build a combined table (ValidPairs) of all trainees from TAO, BAO, and Delegate roles**

VAR ValidPairs =

DISTINCT (

UNION (

**-- TAO mapping**

SELECTCOLUMNS (

'AMH - ROA Applications',

"Trainee", 'AMH - ROA Applications'[TAO],

"Email", 'AMH - ROA Applications'[TAO Email],

"TAO", 'AMH - ROA Applications'[TAO],

"BAO Delegate", BLANK(),

"BAO", BLANK(),

"TAO Delegate", BLANK(),

"TAO 1-UP Manager", 'AMH - ROA Applications'[TAO 1-up Manager],

"TAO 1-UP Manager Email", 'AMH - ROA Applications'[TAO 1-up Manager Email],

"TAO Delegate 1-UP Manager", BLANK(),

"TAO Delegate 1-UP Manager Email", BLANK(),

"BAO 1-UP Manager", BLANK(),

"BAO 1-UP Manager Email", BLANK(),

"BAO Delegate 1-UP Manager", BLANK(),

"BAO Delegate 1-UP Manager Email", BLANK()

),

**-- TAO Delegate mapping**

SELECTCOLUMNS (

'AMH - ROA Applications',

"Trainee", 'AMH - ROA Applications'[TAO Delegate],

"Email", 'AMH - ROA Applications'[TAO Delegate Email],

"TAO", BLANK(),

"BAO", BLANK(),

"BAO Delegate", BLANK(),

"TAO Delegate", 'AMH - ROA Applications'[TAO Delegate],

"TAO 1-UP Manager", BLANK(),

"TAO 1-UP Manager Email", BLANK(),

"TAO Delegate 1-UP Manager", 'AMH - ROA Applications'[TAO Delegate 1-up Manager],

"TAO Delegate 1-UP Manager Email", 'AMH - ROA Applications'[TAO Delegate 1-up Manager Email],

"BAO 1-UP Manager", BLANK(),

"BAO 1-UP Manager Email", BLANK(),

"BAO Delegate 1-UP Manager", BLANK(),

"BAO Delegate 1-UP Manager Email", BLANK()

),

**-- BAO mapping**

SELECTCOLUMNS (

'AMH - ROA Applications',

"Trainee", 'AMH - ROA Applications'[BAO],

"Email", 'AMH - ROA Applications'[BAO Email],

"TAO", BLANK(),

"BAO", 'AMH - ROA Applications'[BAO],

"BAO Delegate", BLANK(),

"TAO Delegate", BLANK(),

"TAO 1-UP Manager", BLANK(),

"TAO 1-UP Manager Email", BLANK(),

"TAO Delegate 1-UP Manager", BLANK(),

"TAO Delegate 1-UP Manager Email", BLANK(),

"BAO 1-UP Manager", 'AMH - ROA Applications'[BAO 1-up Manager],

"BAO 1-UP Manager Email", 'AMH - ROA Applications'[BAO 1-up Manager Email],

"BAO Delegate 1-UP Manager", BLANK(),

"BAO Delegate 1-UP Manager Email", BLANK()

),

**-- BAO Delegate mapping**

SELECTCOLUMNS (

'AMH - ROA Applications',

"Trainee", 'AMH - ROA Applications'[BAO Delegate],

"Email", 'AMH - ROA Applications'[BAO Delegate Email],

"TAO", BLANK(),

"BAO", BLANK(),

"BAO Delegate", 'AMH - ROA Applications'[BAO Delegate],

"TAO Delegate", BLANK(),

"TAO 1-UP Manager", BLANK(),

"TAO 1-UP Manager Email", BLANK(),

"TAO Delegate 1-UP Manager", BLANK(),

"TAO Delegate 1-UP Manager Email", BLANK(),

"BAO 1-UP Manager", BLANK(),

"BAO 1-UP Manager Email", BLANK(),

"BAO Delegate 1-UP Manager", 'AMH - ROA Applications'[BAO Delegate 1-up Manager],

"BAO Delegate 1-UP Manager Email", 'AMH - ROA Applications'[BAO Delegate 1-up Manager Email]

)

)

)

**-- Step 2: Add Meeting 1 attendance status for each trainee by matching email**

VAR BaseTable =

ADDCOLUMNS (

FILTER (ValidPairs, [Trainee] <> BLANK() && [Email] <> BLANK()),

"Meeting 1",

VAR Attended =

MAXX (

FILTER (

'DIAMOND Resource Owner Attestat',

'DIAMOND Resource Owner Attestat'[Email] = EARLIER([Email])

),

'DIAMOND Resource Owner Attestat'[Attendance Status]

)

RETURN IF (Attended = "Yes", "Yes", "No")

)

**-- Step 3: Return the final result combining TAO, TAO Delegate, and BAO roles with application codes**

RETURN

UNION (

**// Output: TAO Records**

SELECTCOLUMNS (

FILTER (BaseTable, NOT(ISBLANK([TAO]))),

"Trainee", [Trainee],

"DIAMOND MAL Code",

CONCATENATEX (

DISTINCT (

FILTER (

'AMH - ROA Applications',

'AMH - ROA Applications'[TAO] = [Trainee]

)

),

'AMH - ROA Applications'[DIAMOND MAL Code], ", "

),

"Email", [Email],

"Role", "TAO",

"1-UP Manager", [TAO 1-UP Manager],

"1-UP Manager Email", [TAO 1-UP Manager Email],

"Meeting 1", [Meeting 1]

),

**// Output: TAO Delegate Records**

SELECTCOLUMNS (

FILTER (BaseTable, NOT(ISBLANK([TAO Delegate]))),

"Trainee", [Trainee],

"DIAMOND MAL Code",

CONCATENATEX (

DISTINCT (

FILTER (

'AMH - ROA Applications',

'AMH - ROA Applications'[TAO Delegate] = [Trainee]

)

),

'AMH - ROA Applications'[DIAMOND MAL Code], ", "

),

"Email", [Email],

"Role", "TAO Delegate",

"1-UP Manager", [TAO Delegate 1-UP Manager],

"1-UP Manager Email", [TAO Delegate 1-UP Manager Email],

"Meeting 1", [Meeting 1]

),

**// Output: BAO Records**

SELECTCOLUMNS (

FILTER (BaseTable, NOT(ISBLANK([BAO]))),

"Trainee", [Trainee],

"DIAMOND MAL Code",

CONCATENATEX (

DISTINCT (

FILTER (

'AMH - ROA Applications',

'AMH - ROA Applications'[BAO] = [Trainee]

)

),

'AMH - ROA Applications'[DIAMOND MAL Code], ", "

),

"Email", [Email],

"Role", "BAO",

"1-UP Manager", [BAO 1-UP Manager],

"1-UP Manager Email", [BAO 1-UP Manager Email],

"Meeting 1", [Meeting 1]

)

)

## DAX Table: **Final Training Progress Summary**

Final Training Progress Summary =

**-- Step 1: Combine rows from both Meeting 1 and Meeting 2 sources into one unified table**

VAR AllRows =

UNION (

**-- Data from Training Progress Tracking (Meeting 1 info)**

SELECTCOLUMNS (

'Training Progress Tracking',

"Trainee", [Trainee],

"Email", [Email],

"Role", [Role],

"DIAMOND MAL Code", [DIAMOND MAL Code],

"1-UP Manager Email", [1-UP Manager Email],

"1-UP Manager", [1-UP Manager],

"Meeting 1", [Meeting 1], -- actual Meeting 1 value

"Meeting 2", BLANK() -- placeholder for Meeting 2

),

**-- Data from Training Progress Tracking 1 (Meeting 2 info)**

SELECTCOLUMNS (

'Training Progress Tracking 1',

"Trainee", [Trainee],

"Email", [Email],

"Role", [Role],

"DIAMOND MAL Code", [DIAMOND MAL Code],

"1-UP Manager Email", [1-UP Manager Email],

"1-UP Manager", [1-UP Manager],

"Meeting 1", BLANK(), -- placeholder for Meeting 1

"Meeting 2", [Meeting 2] -- actual Meeting 2 value

)

)

**-- Step 2: Group data by trainee, email, role, etc., and retrieve their actual Meeting 1 & Meeting 2 values**

VAR FinalTable =

ADDCOLUMNS (

**-- Remove duplicates and group by participant details**

SUMMARIZE (

AllRows,

[Trainee],

[Email],

[Role],

[DIAMOND MAL Code],

[1-UP Manager Email],

[1-UP Manager]

),

**-- Get Meeting 1 status using MAXX from filtered AllRows**

"Meeting 1",

MAXX (

FILTER (

AllRows,

[Trainee] = EARLIER([Trainee]) &&

[Email] = EARLIER([Email]) &&

[Role] = EARLIER([Role])

),

[Meeting 1]

),

**-- Get Meeting 2 status using MAXX from filtered AllRows**

"Meeting 2",

MAXX (

FILTER (

AllRows,

[Trainee] = EARLIER([Trainee]) &&

[Email] = EARLIER([Email]) &&

[Role] = EARLIER([Role])

),

[Meeting 2]

),

**-- Step 3: Create a Final Meeting Status column**

**-- If either Meeting 1 or Meeting 2 is "Yes", mark "Yes"; otherwise "No"**

"Final Meeting Status",

VAR M1 =

MAXX (

FILTER (

AllRows,

[Trainee] = EARLIER([Trainee]) &&

[Email] = EARLIER([Email]) &&

[Role] = EARLIER([Role])

),

[Meeting 1]

)

VAR M2 =

MAXX (

FILTER (

AllRows,

[Trainee] = EARLIER([Trainee]) &&

[Email] = EARLIER([Email]) &&

[Role] = EARLIER([Role])

),

[Meeting 2]

)

RETURN

IF (M1 = "Yes" || M2 = "Yes", "Yes", "No")

)