

Power BI Row Level Security for University Data

TAIR Annual Conference

March 1, 2022

Faron Kincheloe | Baylor University

Faron_Kincheloe@baylor.edu

Objectives

- Define Row Level Security
- Explore University Scenarios
 - Basic
 - Intermediate
 - Advanced
 - Pushing the Envelope

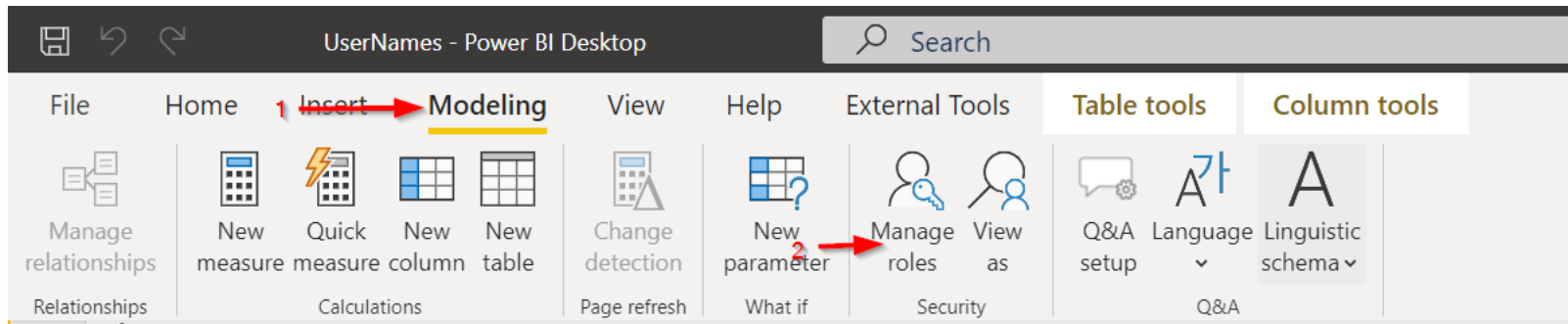
Disclaimers

- I don't know everything about Row Level Security
- Assumes some knowledge of Power BI development
- Involves DAX coding and possibly programming depending on complexity
- Context is primarily in the Power BI Service (cloud)

What is Row Level Security (RLS)?

- A way to restrict access to data based on logged in user
- “Automatic” filtering based on ID
- Not used on every dashboard
- Teams, channels and workspaces limit most access
- RLS allows different users to access different data within a single shared report – prevents need to maintain multiple copies

Basic RLS Configuration



1. Modeling tab
2. Manage roles

Basic RLS Configuration

Manage roles

Roles

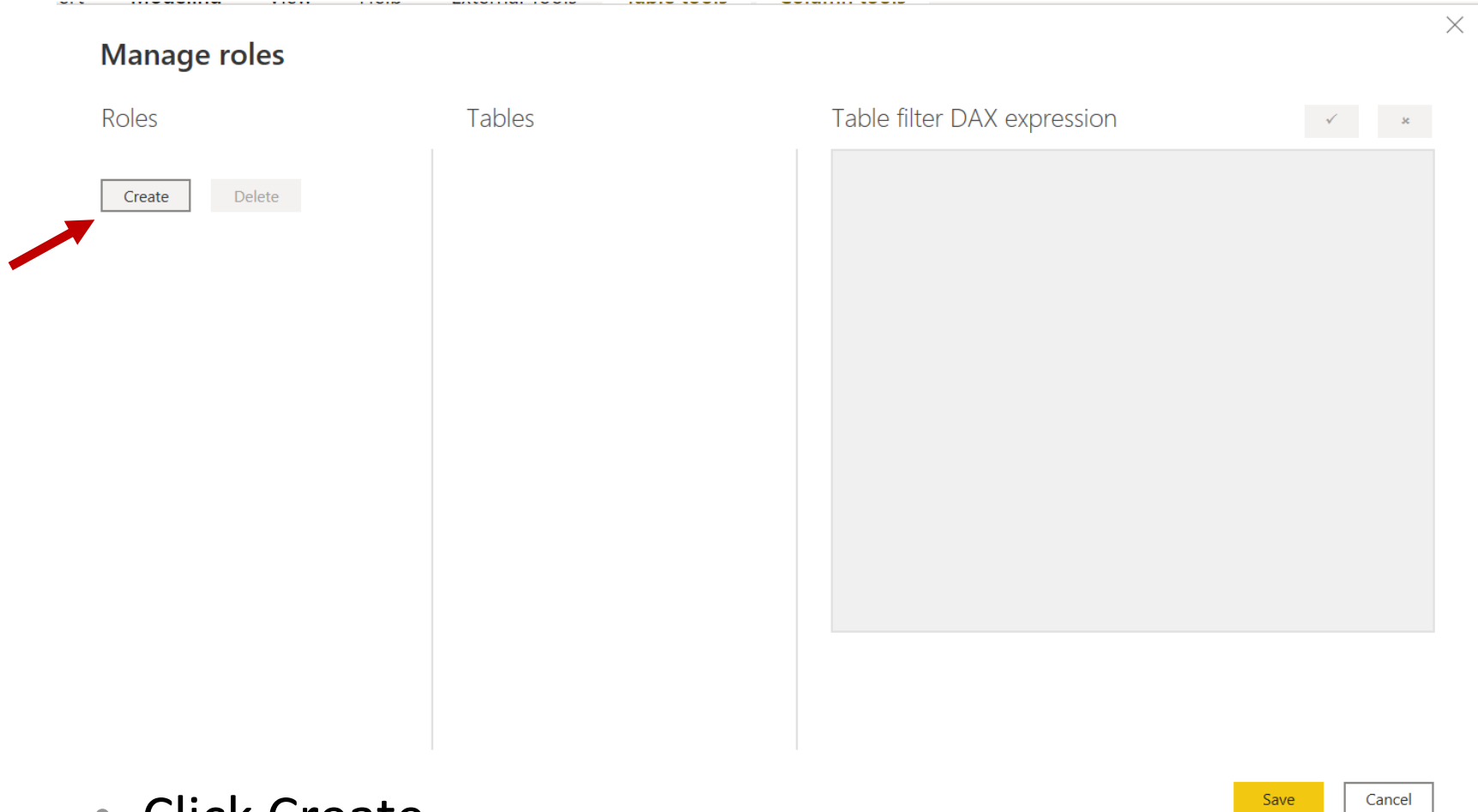
CreateDelete

Tables

Table filter DAX expression

✓✕

SaveCancel



• Click Create

Save

Cancel

Basic RLS Configuration

Manage roles

The screenshot shows the 'Manage roles' interface. On the left, the 'Roles' section has a table with one row: 'Biology'. Below it are 'Create' and 'Delete' buttons. On the right, the 'Tables' section has a table with one row: 'Table'. A red arrow points to the ellipsis on the right of the 'Table' row. To the right of the 'Tables' section is a 'Table filter DAX expression' area with a large text input box and '✓' and '✕' buttons. Below the input box is a note: 'Filter the data that this role can see by entering a DAX filter expression that returns a True/False value. For example: [Entity ID] = "Value"'.

1. Give the Role a name

2. Click ellipses on table

- Click Add Filter
- Select column

Basic RLS Configuration

Manage roles

Roles

Biology ...

Create

Delete

Tables

Table ...

Table filter DAX expression



[Department] = "Value"

Filter the data that this role can see by entering a DAX filter expression that returns a True/False value. For example: [Entity ID] = "Value"

- Power BI auto populates

Basic RLS Configuration

Manage roles

Roles

Biology ...

Create

Delete

Tables

Table ...

Table filter DAX expression

✓

✗

[Department] = "BIO"

Filter the data that this role can see by entering a DAX filter expression that returns a True/False value. For example: [Entity ID] = "Value"

- Change to desired value

Basic RLS Configuration

Manage roles

The screenshot displays the 'Manage roles' interface with three main sections: Roles, Tables, and Table filter DAX expression.

- Roles:** A list containing 'Biology' and 'Chemistry'. Below the list are 'Create' and 'Delete' buttons. A red arrow labeled '1' points to the 'Create' button.
- Tables:** A list containing 'Table'. A red arrow labeled '2' points to the 'Table' entry.
- Table filter DAX expression:** A text area containing the expression `[Department] = "CHE"`. Above the text area are two buttons: a checkmark (✓) and a cancel (✕). A red arrow labeled '2' points to the checkmark button.

Below the DAX expression text area, there is a descriptive text: "Filter the data that this role can see by entering a DAX filter expression that returns a True/False value. For example: [Entity ID] = "Value"". At the bottom right, there are 'Save' and 'Cancel' buttons. A red arrow labeled '3' points to the 'Save' button.

1. Create all the roles you need
2. Check syntax
3. Save when completed

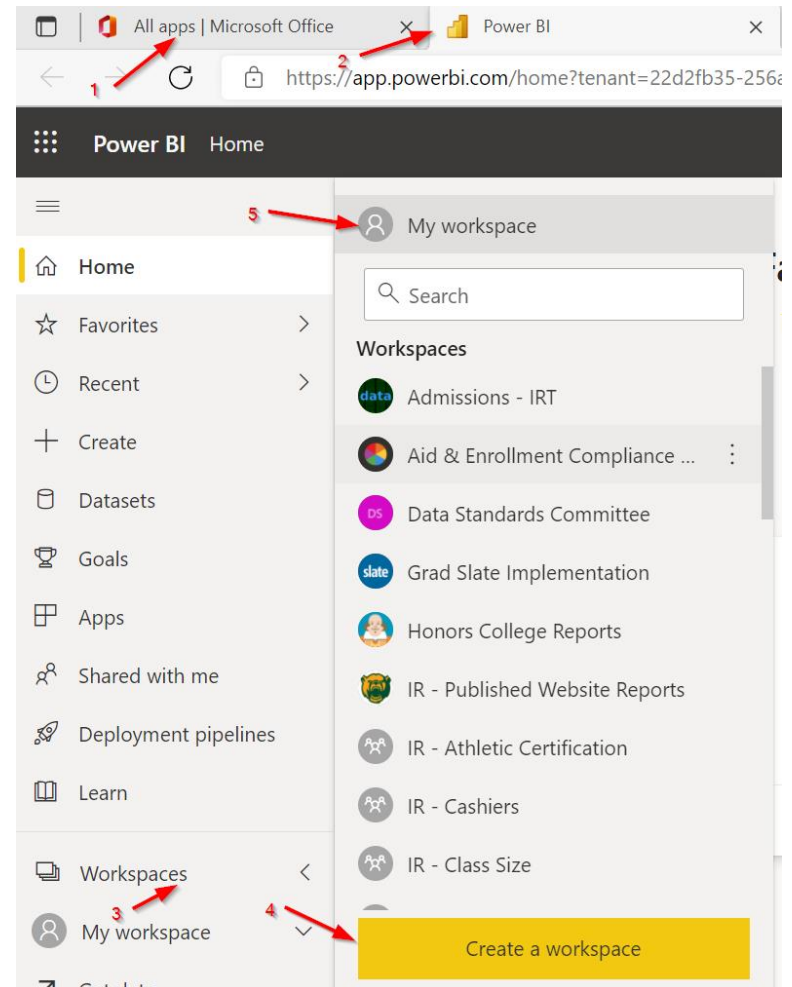
Basic RLS Configuration

The screenshot shows the Power BI Desktop interface with the 'Home' tab selected. The 'Publish' button is highlighted in the 'Share' group. A 'Publish to Power BI' dialog box is open, showing a list of workspaces. 'My workspace' is selected. The 'Select' button is highlighted.

1. Home tab
2. Publish
3. Choose a workspace
4. Click Select

Basic RLS Configuration

1. Login office.com
2. Select Power BI App
3. Click Workspaces
4. Add new workspaces if needed (optional)
5. Select workspace where dashboard was published



Basic RLS Configuration

1. Click dots by Dataset
2. Select Security

My workspace

+ New ▾

All Content Datasets + dataflows

	Name	
	Law Degrees	
	Law Degrees	
	UG Recruiting List-Personalize	
	UG Recruiting List-Personalize	
	UserNames	
	UserNames	

1

2

- Analyze in Excel
- Create report
- Create paginated report
- Delete
- Quick insights
- Security
- Rename
- Settings
- Download this file
- Manage permissions
- View lineage

Dataset Fa

Basic RLS Configuration

1. Select Role
2. Enter a user or group
3. Click Add
4. Click Save when finished adding

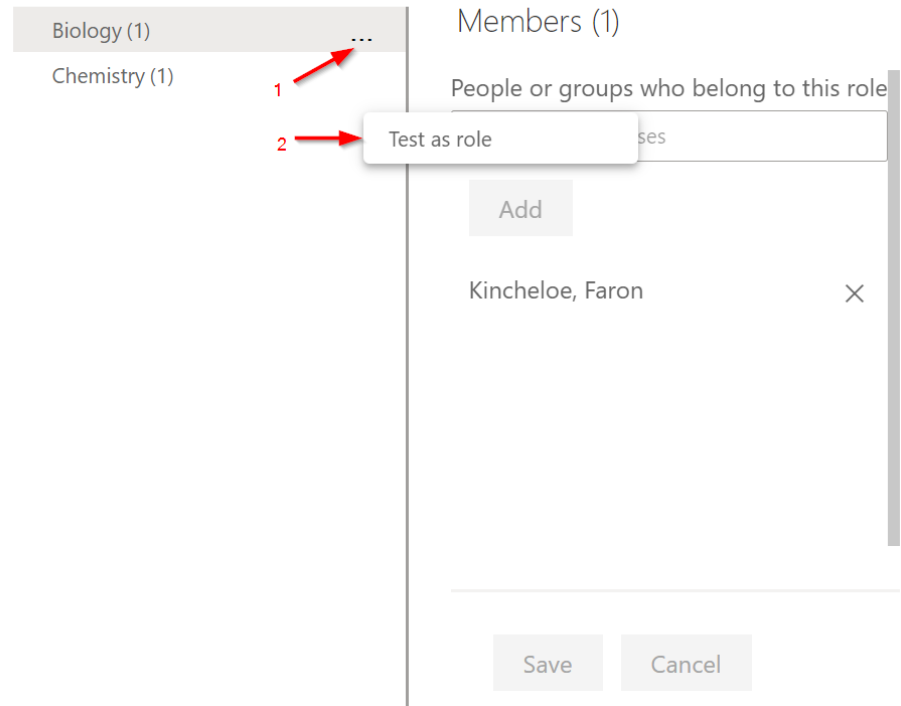
Row-Level Security

The screenshot displays the 'Row-Level Security' configuration window. On the left, under 'Biology (1)', the 'Chemistry (0)' role is selected, indicated by a red arrow labeled '1'. On the right, the 'Members (0)' section shows a list of people or groups. A red arrow labeled '2' points to the 'Enter email addresses' input field. A red arrow labeled '3' points to the yellow 'Add' button. At the bottom, a red arrow labeled '4' points to the 'Save' button. The interface also includes a 'Cancel' button and a list of existing members, currently showing 'Kincheloe, Faron' with a remove icon.

Testing RLS Configuration

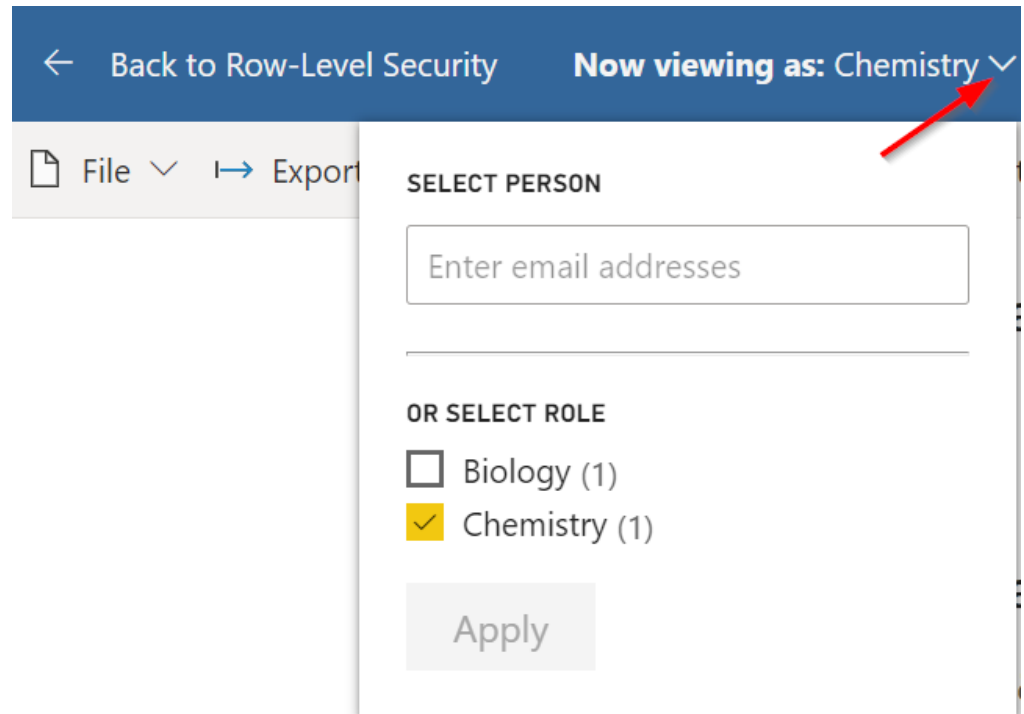
1. Click ellipses beside role
2. Click Test as role

Row-Level Security



Testing RLS Configuration

- Click dropdown to change role or test a specific user



The screenshot shows a web interface for testing Row-Level Security (RLS) configuration. The top navigation bar is blue and contains a back arrow, the text "Back to Row-Level Security", and a dropdown menu labeled "Now viewing as: Chemistry". A red arrow points to the dropdown arrow. Below the navigation bar, there is a light gray bar with "File" and "Export" options. The main content area is a white modal window titled "SELECT PERSON" with a text input field labeled "Enter email addresses". Below this, there is a section titled "OR SELECT ROLE" with two radio button options: "Biology (1)" and "Chemistry (1)". The "Chemistry (1)" option is selected, indicated by a yellow checkmark. At the bottom of the modal is a gray "Apply" button.

← Back to Row-Level Security **Now viewing as: Chemistry** ▾

File ▾ Export

SELECT PERSON

Enter email addresses

OR SELECT ROLE

☐ Biology (1)

☒ Chemistry (1)

Apply

Dynamic RLS

- If only I didn't have to maintain so many roles...
- Is there a way to use the University org chart?
- Use dynamic security with an access table!!!
- At least two tables.
 - One or more for data
 - One with access levels
 - Org chart (in your dreams)
 - Custom access table(reality)

Dynamic RLS

- PowerBI DAX functions for RLS
- USERNAME()
- USERPRINCIPALNAME()
- LOOKUPVALUE()
- PATH()
- PATHCONTAINS()
- CONCATENATE()

Dynamic RLS

- PowerBI DAX functions for RLS
- USERNAME()
 - The user name from credentials given at connection time.
 - May be advantages for shared desktop files
- USERPRINCIPALNAME()
 - Typically the user email (depends on system)
 - Preferred for Power BI Service environment

Personal Computer

WINDOWS-I0A82TG\kinchelf
UserName

WINDOWS-I0A82TG\kinchelf
UserPrincipalName

University Domain Workstation

BAYLOR\Faron_Kincheloe
UserName

Faron_Kincheloe@baylor.edu
UserPrincipalName

Powerbi.com

Faron_Kincheloe@baylor.edu

UserName

Faron_Kincheloe@baylor.edu

UserPrincipalName

Dynamic RLS

- LOOKUPVALUE()
 - Returns the value for the row that meets all criteria specified by one or more search conditions.

```
LOOKUPVALUE(  
    <result_columnName>,  
    <search_columnName>,  
    <search_value>  
    [, <search2_columnName>,  
    <search2_value>]...  
    [, <alternateResult>]  
)
```

Dynamic RLS

- LOOKUPVALUE() - Example

Bear_ID	PBI_level
A_M@baylor.edu	MLC
Amanda_D@baylor.edu	BIO
A_H@baylor.edu	HH
B_H@baylor.edu	HIS
B_Bell@baylor.edu	REL
B_B@baylor.edu	AS
B_R@baylor.edu	MTH
Carl_L@baylor.edu	SWO
C_C@baylor.edu	ALL

Table filter DAX expression

```
var CurrentLoggedIn =  
    LOOKUPVALUE(RLS_access[PBI_level],  
                RLS_access[Bear_ID],  
                USERPRINCIPALNAME()  
    )  
RETURN
```


Dynamic RLS

- `PATH()`
 - Builds a parent-child hierarchy across related rows of a table.
 - Used in tables that have some kind of internal hierarchy (employee data).
 - Did not help with student data.

```
ORG_Path = PATH(Employee[EmployeeKey],  
Employee[ManagerKey])
```

Dynamic RLS

- PATH() - Example

Person	EmployeeKey	ManagerKey	ORG_path
President	1000		1000
Provost	1100	1000	1000 1100
Bus. Dean	1200	1100	1000 1100 1200

Dynamic RLS

- When “out of the box” doesn’t work – build your own.
- CONCATENATE()
 - Joins two text strings into one text string.
 - Uses literals or column references
 - Can be nested to join multiple values

CONCATENATE(<text1>, <text2>)

Dynamic RLS – Building a Path

1. Select Data icon and choose desired data table if not already selected.
2. Select new column.

The screenshot shows the Power BI Desktop interface. The title bar reads 'RLSTest - Power BI Desktop'. The ribbon is set to 'Table tools'. The 'Name' field shows 'ug_recruits'. The 'Table tools' ribbon has several groups: 'Structure' (with a 'Name' field), 'Calendars' (with 'Mark as date table'), 'Relationships' (with 'Manage relationships'), and 'Calculations' (with 'New measure', 'Quick measure', 'New column', and 'New table'). A red arrow labeled '2' points to the 'New column' icon. In the bottom-left corner, a red arrow labeled '1' points to the 'Data' icon in the 'Structure' group. Below the ribbon, a data table is displayed with the following columns and data:

APP_TERM	PROSPECT_LEVEL	SLATE_REF_ID	LAST_NAME	FIRST_NAME	MIDDLE_NAME
Fall 2021	F	893182727	Pulley	Freighton	Ronan
Fall 2021	F	91781905	Schubert	Benjamin	Charles
Fall 2021	F	686302029	Garcia	Julian	Oliver

Dynamic RLS – Building a Path

```
ORG_Path = CONCATENATE(  
    CONCATENATE("ALL|", ug_recruits[COLLEGE]),  
    CONCATENATE("|", ug_recruits[DEPT1]))  
ORG_Path = ALL|HH|FCS
```

The screenshot shows the Power BI Desktop interface with the 'Column tools' ribbon selected. The formula bar displays the DAX formula for the 'ORG_Path' column:

```
1 ORG_Path = CONCATENATE(CONCATENATE("ALL|",ug_recruits[COLLEGE]),CONCATENATE("|",ug_recruits[DEPT1]))
```

The formula is highlighted with a red circle. Below the formula bar, a table of data is displayed, also with the 'ORG_Path' column highlighted by a red circle. The table has the following columns: _STATUS, LINE_CAMP_TYPE, LINE_CAMP_REG_IND, LINE_CAMP_STATUS, VA_BENEFITS, LEVEL_YEAR, NET_DEPOSIT_IND, ENROLLED_IND, PBI_ROW, and ORG_Path.

_STATUS	LINE_CAMP_TYPE	LINE_CAMP_REG_IND	LINE_CAMP_STATUS	VA_BENEFITS	LEVEL_YEAR	NET_DEPOSIT_IND	ENROLLED_IND	PBI_ROW	ORG_Path
					FR21	0	0	292980	ALL HH FCS
					FR21	0	0	101640	ALL EN CSI
					FR21	0	0	278674	ALL HH HHPR
					FR21	0	0	368400	ALL HH HHPR
					FR21	0	0	103450	ALL NU NUR
					FR21	0	0	389991	ALL HH CSD

Dynamic RLS

- Completing the puzzle!
- PATHCONTAINS()
 - Returns TRUE if the specified item exists within the specified path.
 - Compare looked up value with path on each row.

PATHCONTAINS(<path>, <item>)

Dynamic RLS

- Completing the puzzle!

PATHCONTAINS(ug_recruits[ORG_Path],CurrentLoggedIn)

Bear_ID	PBI_level
A_M@baylor.edu	MLC
Amanda_D@baylor.edu	BIO
A_H@baylor.edu	HH
Carl_L@baylor.edu	SWO
C_C@baylor.edu	ALL

ORG_Path	Filter result
ALL HH FCS	TRUE
ALL EN CSI	FALSE

Table filter DAX expression ☒

```
var CurrentLoggedIn =  
    LOOKUPVALUE(RLS_access[PBI_level],  
                RLS_access[Bear_ID],  
                USERPRINCIPALNAME()  
    )  
  
RETURN  
    PATHCONTAINS(ug_recruits[ORG_Path],CurrentLoggedIn)
```

Dynamic RLS

- Completing the puzzle!

PATHCONTAINS(ug_recruits[ORG_Path],CurrentLoggedIn)

Bear_ID	PBI_level
A_M@baylor.edu	MLC
Amanda_D@baylor.edu	BIO
A_H@baylor.edu	HH
Carl_L@baylor.edu	SWO
C_C@baylor.edu	ALL

ORG_Path	Filter result
ALL HH FCS	TRUE
ALL EN CSI	TRUE

Table filter DAX expression ☒

```
var CurrentLoggedIn =  
    LOOKUPVALUE(RLS_access[PBI_level],  
                RLS_access[Bear_ID],  
                USERPRINCIPALNAME()  
                )  
  
RETURN  
    PATHCONTAINS(ug_recruits[ORG_Path],CurrentLoggedIn)
```


Dynamic RLS

- Access table can be maintained within the PowerBI report or imported from a separate table
- Most of our reports have scheduled refresh through Power Automate

Dynamic RLS

- Security role is applied to all Faculty and Staff

The screenshot displays the Power BI interface for configuring Row-Level Security (RLS). The top navigation bar shows the path: Power BI > IR - UG Recruiting > Row-Level Security. On the left, a sidebar menu includes options like Home, Favorites, Recent, Create, Datasets, Apps, Shared with me, and Learn. The main content area is titled 'Row-Level Security' and features a list of security roles. The role 'Dynamic Security (1)' is selected and highlighted. To the right of this list, the 'Members (1)' section shows the role is assigned to 'Faculty/Staff'. Below this, there is a text input field labeled 'Enter email addresses' and an 'Add' button. A close button (X) is located next to the 'Faculty/Staff' role name.

Power BI IR - UG Recruiting > Row-Level Security

Row-Level Security

Dynamic Security (1)

Members (1)

People or groups who belong to this role

Enter email addresses

Add

Faculty/Staff X

Dynamic RLS - Intermediate

- What if someone needs two or more departments?
- Switch path and item
- COLLEGE and DEPT1 should not share any values

```
(PATHCONTAINS(CurrentLoggedIn,ug_recruits[DEPT1])  
|| PATHCONTAINS(CurrentLoggedIn,ug_recruits[COLLEGE])  
|| CurrentLoggedIn="ALL")
```

Dynamic RLS

- What if someone needs two or more departments?
- Switch path and item

Bear_ID	PBI_level
A_M@baylor.edu	CSI
Amanda_D@baylor.edu	HH EN
A_H@baylor.edu	HH
Carl_L@baylor.edu	SWO
C_C@baylor.edu	ALL

College	Dept1	Filter result
HH	FCS	
EN	CSI	

Table filter DAX expression



```
var CurrentLoggedIn =  
    LOOKUPVALUE(RLS_access[PBI_level],  
                RLS_access[Bear_ID],  
                USERPRINCIPALNAME()  
    )  
  
RETURN  
    (PATHCONTAINS(CurrentLoggedIn,ug_recruits[DEPT1])  
    || PATHCONTAINS(CurrentLoggedIn,ug_recruits[COLLEGE])  
    || CurrentLoggedIn="ALL")
```

Dynamic RLS

- What if someone needs two or more departments?
- Switch path and item

Bear_ID	PBI_level
A_M@baylor.edu	CSI
Amanda_D@baylor.edu	HH EN
A_H@baylor.edu	HH
Carl_L@baylor.edu	SWO
C_C@baylor.edu	ALL

College	Dept1	Filter result
HH	FCS	FALSE
EN	CSI	TRUE

Table filter DAX expression

```
var CurrentLoggedIn =  
    LOOKUPVALUE(RLS_access[PBI_level],  
                RLS_access[Bear_ID],  
                USERPRINCIPALNAME()  
            )  
  
RETURN  
    (PATHCONTAINS(CurrentLoggedIn,ug_recruits[DEPT1])  
     || PATHCONTAINS(CurrentLoggedIn,ug_recruits[COLLEGE])  
     || CurrentLoggedIn="ALL")
```

Dynamic RLS

- What if someone needs two or more departments?
- Switch path and item

Bear_ID	PBI_level
A_M@baylor.edu	MLC
Amanda_D@baylor.edu	HH EN
A_H@baylor.edu	HH
Carl_L@baylor.edu	SWO
C_C@baylor.edu	ALL

College	Dept1	Filter result
HH	FCS	TRUE
EN	CSI	TRUE

Table filter DAX expression

```
var CurrentLoggedIn =  
    LOOKUPVALUE(RLS_access[PBI_level],  
                RLS_access[Bear_ID],  
                USERPRINCIPALNAME()  
            )  
  
RETURN  
    (PATHCONTAINS(CurrentLoggedIn,ug_recruits[DEPT1])  
    || PATHCONTAINS(CurrentLoggedIn,ug_recruits[COLLEGE])  
    || CurrentLoggedIn="ALL")
```

Dynamic RLS

- What if someone needs two or more departments?
- Switch path and item

Bear_ID	PBI_level
A_M@baylor.edu	CSI
Amanda_D@baylor.edu	HH EN
A_H@baylor.edu	HH
Carl_L@baylor.edu	SWO
C_C@baylor.edu	ALL

College	Dept1	Filter result
HH	FCS	TRUE
EN	CSI	FALSE

Table filter DAX expression

```
var CurrentLoggedIn =  
    LOOKUPVALUE(RLS_access[PBI_level],  
                RLS_access[Bear_ID],  
                USERPRINCIPALNAME()  
            )  
  
RETURN  
    (PATHCONTAINS(CurrentLoggedIn,ug_recruits[DEPT1])  
     || PATHCONTAINS(CurrentLoggedIn,ug_recruits[COLLEGE])  
     || CurrentLoggedIn="ALL")
```

Dynamic RLS

- What if someone needs two or more departments?
- Switch path and item

Bear_ID	PBI_level
A_M@baylor.edu	CSI
Amanda_D@baylor.edu	HH EN
A_H@baylor.edu	HH
Carl_L@baylor.edu	SWO
C_C@baylor.edu	ALL

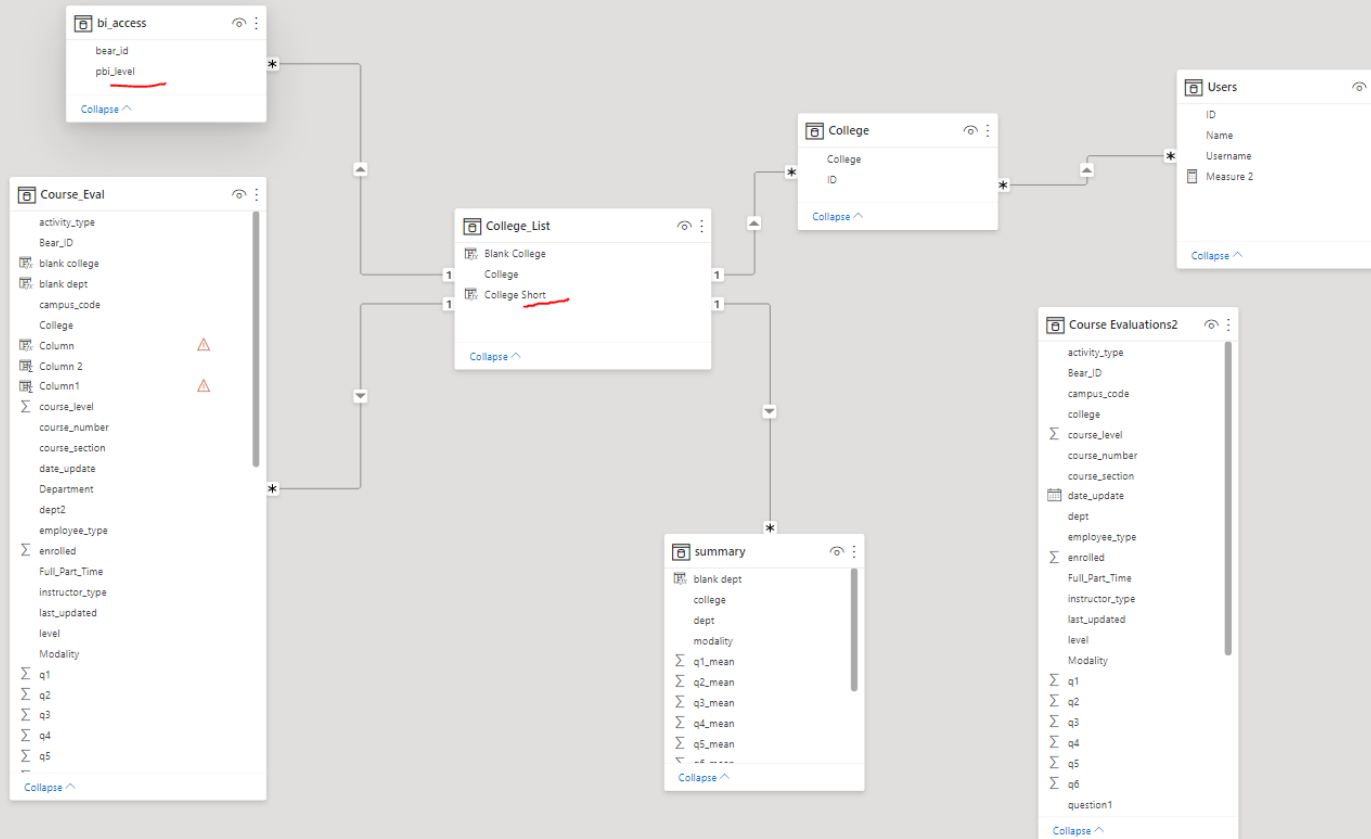
College	Dept1	Filter result
HH	FCS	TRUE
EN	CSI	TRUE

Table filter DAX expression

```
var CurrentLoggedIn =  
    LOOKUPVALUE(RLS_access[PBI_level],  
                RLS_access[Bear_ID],  
                USERPRINCIPALNAME()  
            )  
  
RETURN  
    (PATHCONTAINS(CurrentLoggedIn,ug_recruits[DEPT1])  
     || PATHCONTAINS(CurrentLoggedIn,ug_recruits[COLLEGE])  
     || CurrentLoggedIn="ALL")
```


Dynamic RLS - Advanced

- Multiple tables, multiple levels



Dynamic RLS - Advanced

- Multiple tables, multiple levels

Bear_ID	PBI_level
A_M@baylor.edu	AS MLC
Amanda_D@baylor.edu	HH DEPTALL
A_H@baylor.edu	HH PMHH MH
B_H@baylor.edu	ALL
B_Bell@baylor.edu	AS ART CSS MLC REL

Dynamic RLS - Advanced

- Multiple tables, multiple levels – two filters

Manage roles

Roles

Security ...

Create Delete

Tables

bi_access	...
College	...
College_List	...
Course Evaluations2	...
Course_Eval	...
summary	...
Users	...

Dynamic RLS - Advanced

- First filter on College_List to lower level users from seeing unauthorized colleges

```
(PATHCONTAINS(CurrentLoggedIn,College_List[College Short])  
|| CurrentLoggedIn="ALL")
```

Dynamic RLS - Advanced

- Second filter on Course_Eval to select rows from data table
- College is not needed in logic below because table relationships in model inherit first filter.

```
(PATHCONTAINS(CurrentLoggedIn,Course_Eval[dept2])
```

```
|| CurrentLoggedIn="ALL"
```

```
|| PATHCONTAINS(CurrentLoggedIn,"DEPTALL"))
```

Pushing the Envelope

Approximately 80 different filter expressions (roles):

LOGIC:DEPT1 not in ('HSD' 'MUS' 'THEA' 'NUR' 'CSI' 'PSYN' 'BIO' 'CHEB' 'PHY' 'ENV' 'GEO' 'IAS') and MAJOR not in ('SRF' 'HSS') and COLLEGE ne 'EN' and ED_GOALS_PRIMARY in (' ' 'PL') and HIGH_SCHOOL_GPA_RECALC ge 3 and PROSPECT_LEVEL='F'

LOGIC:ED_LEVEL in ('PL') or MAJOR='PLAW'

LOGIC:FGCS_IND=1

LOGIC:PROSPECT_LEVEL='T' or FGCS_IND=1

LOGIC:college='BU' and High_Ability='Y'

LOGIC:ed_goals_primary IN('DA' 'PA' 'PC' 'PD' 'PF' 'PH' 'PL' 'PM' 'PO' 'PP' 'PT' 'PV')

LOGIC:ed_goals_primary IN('DA' 'PA' 'PC' 'PD' 'PF' 'PH' 'PL' 'PM' 'PO' 'PP' 'PT' 'PV') or major in('BIO' 'PBIO' 'CHE' 'BCHE' 'PSY' 'NSC' 'MH' 'HSS' 'NUR' 'NUTN' 'CSD' 'CHEA' 'PHY' 'ENSC')

Pushing the Envelope

- Borrow the concept of indexing from R, Excel, Bingo, and Battleship
- Build security and indexing into data tables outside of PowerBI
 1. Add a PBI_Row column with serialized row numbers to the data table.
 2. Create access table with list of allowed rows for each user.
 - A. Use filter expression in where clause of query
 - B. Return User Name and row numbers
 - C. Concatenate all records together

Pushing the Envelope

- Data table and access table must be refreshed together every time
- Additional options in LOOKUPVALUE function tie two tables together:

[PBI_ROW] =

LOOKUPVALUE(

'access_table'[PBI_Row],

'access_table' [Bear_ID] , UserPrincipalName(),

'access_table' [PBI_Row],

'data_table'[PBI_Row])

Pushing the Envelope

Manage roles

Roles

Dynamic Security

Create

Delete

Tables

ug_recruits

ug_recruits_access

Table filter DAX expression

```
[PBI_ROW] =  
LOOKUPVALUE(  
  'ug_recruits_access'[PBI_Row],  
  'ug_recruits_access'[Bear_ID] , UserPrincipalName(),  
  'ug_recruits_access'[PBI_Row],  
  'ug_recruits'[PBI_Row]  
)
```

Filter the data that this role can see by entering a DAX filter expression that returns a True/False value. For example: [Entity ID] = "Value"

Save

Cancel