

BI4Dynamics – Setting up Roles & Permissions in Tabular Database

Last update: March 2023
Version: 1.5

Table of Contents

| | | |
|-----|--|----|
| 1 | INTRODUCTION..... | 2 |
| 1.1 | About Security | 2 |
| 1.2 | Tabular Database Implementation Options | 2 |
| 1.3 | Microsoft documentation..... | 3 |
| 2 | SETTING UP ROLES AND PERMISSIONS..... | 4 |
| 2.1 | Connect to Tabular database with SSMS..... | 4 |
| 2.2 | Create roles..... | 4 |
| 2.3 | Add Membership (users or user groups)..... | 5 |
| 2.4 | Add DAX expression into DAX Filter | 5 |
| 2.5 | Obtain the correct name of the desired filtering column | 6 |
| 2.6 | Check the syntax of DAX Filters | 8 |
| 3 | SETTING UP DYNAMIC ROLES..... | 9 |
| 3.1 | Identifying connected user | 9 |
| 3.2 | Bridge table..... | 9 |
| 3.3 | DAX expression for Dynamic security | 9 |
| 4 | OPTION B: SETTING IN AZURE PORTAL WHEN USING AZURE ANALYSIS SERVICES | 11 |
| 5 | OPTION C: SETTING IN POWER BI SERVICE WHEN USING POWER BI PREMIUM..... | 12 |
| 6 | SETTING UP ROLES WITH OBJECT-LEVEL SECURITY USING TABULAR EDITOR..... | 13 |

1 INTRODUCTION

Implementation services cover standard software installation with configuration and reports connection.

1.1 About Security

Security is a feature of **all Microsoft's** developed and maintained **programs**:

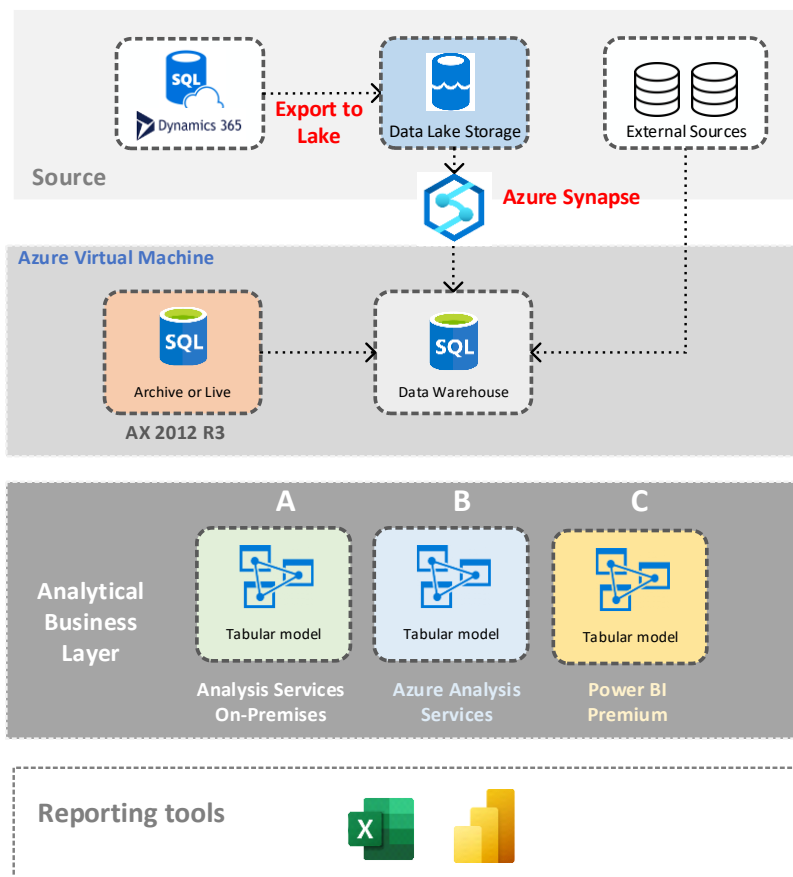
- **BI4Dynamics** provides **content**
- **Microsoft** provides **security**

All of the services have **identical content**, but **different security management**.

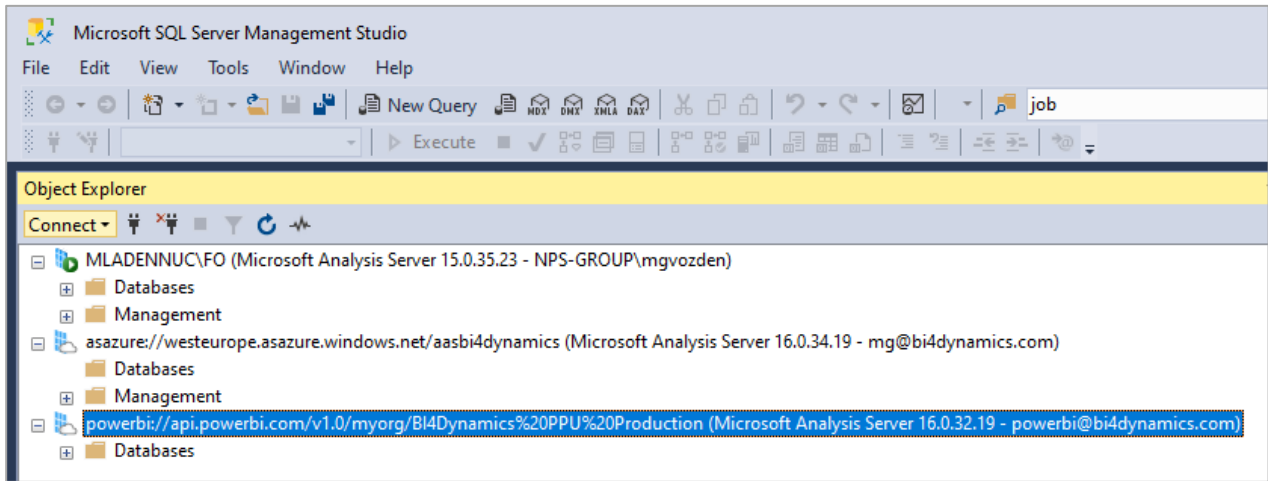
1.2 Tabular Database Implementation Options

BI4Dynamics publishes data from **data warehouse** to **tabular model** that can be hosted:

- **Locally (On-Premises)** or in **Azure VM** as part of SQL Server
- By **Microsoft in Azure** as **Azure Analysis Services**
- By **Microsoft in Power BI Premium Service** as **XMLA capacity**



Tabular database can be access by **SSMS** (SQL Server Management Studio):



1.3 Microsoft documentation

| |
|---|
| URL |
| Tabular modeling – Roles |
| Create and manage roles with SSMS |

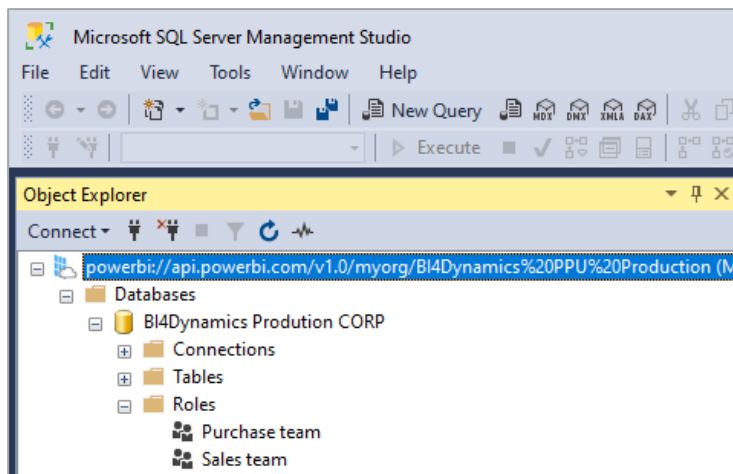
2 SETTING UP ROLES AND PERMISSIONS

This chapter **shows how to**: connect tabular database, create roles, add membership, obtain the correct name of a column, add DAX expressions into DAX Filter and check the syntax of it.

2.1 Connect to Tabular database with SSMS

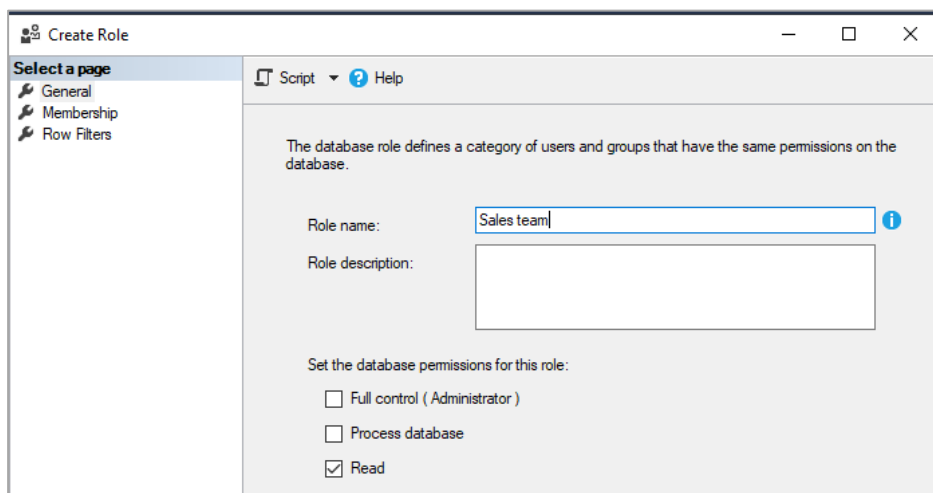
Connecting to any type of Tabular database with SSMS is the same.

- Select **Server / Workspace**
- Select **Tabular Database**

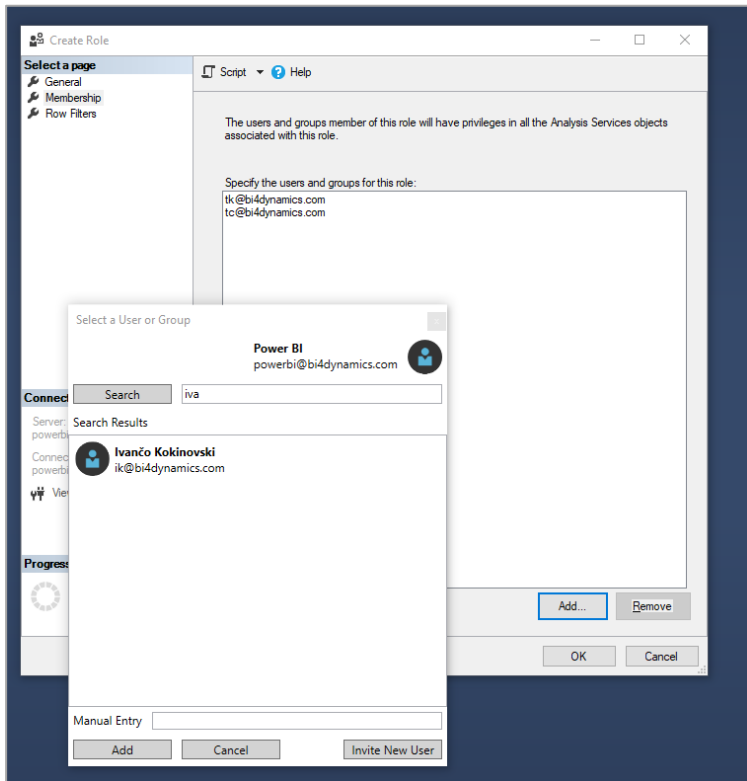


2.2 Create roles

Admin user can create a new Role. Go to Roles and click **"New Role"**



2.3 Add Membership (users or user groups)



Local service requires **Windows login** from **local domain**, **Azure** and **Power BI** get users from **AAD** (Azure Active Directory). It is recommended to **save the role** and **reopen it**.

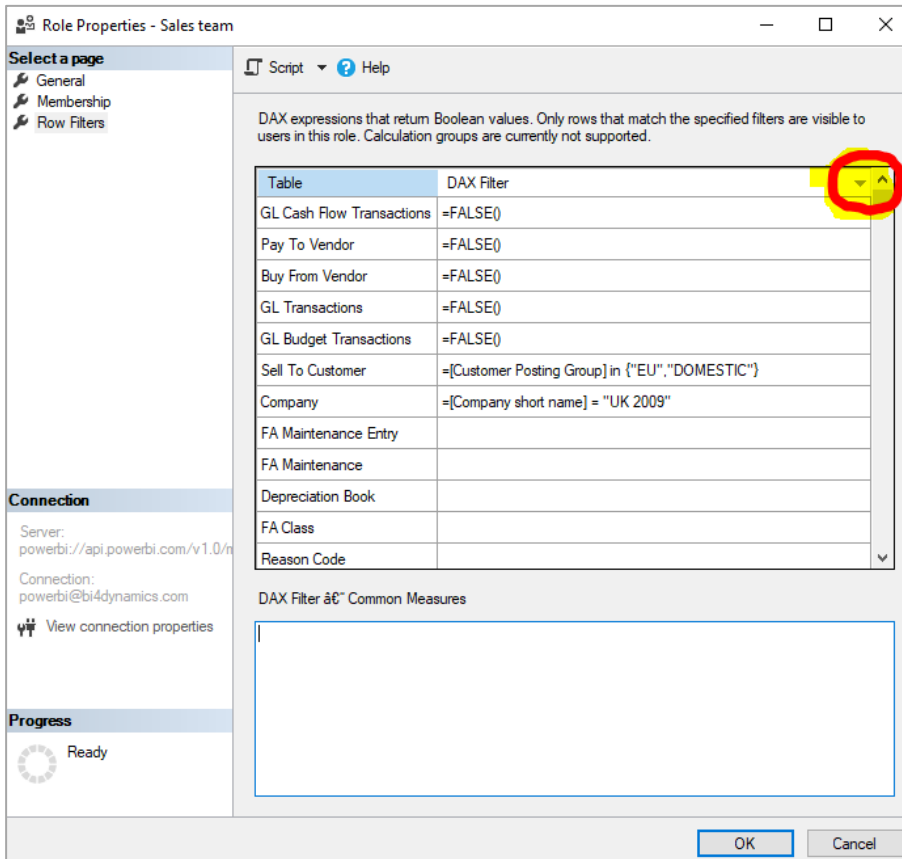
2.4 Add DAX expression into DAX Filter

Select the desired filtering table and **add DAX expression** into **DAX Filter column**.

DAX Filter examples:

| Table in Tabular Database | Table Type | DAX Filter expression | Comment |
|---------------------------|-------------|--|--|
| GL Transaction | Transaction | =FALSE() | Access to transactional table is denied |
| Pay to Vendor | Dimension | =FALSE() | Access to dimensional and all related transactional tables is denied (Purchase and Payables) |
| Sell To Customer | Dimension | = [Customer Posting Group] IN {"EU","DOMESTIC"} | Allowed only filtered rows and all related transactions |
| Company | Dimension | = [Company Code] IN {"E001","E004","E005"} | Allowed only filtered rows and all related transactions |
| Item | Dimension | = [Sales Category 2 Name]="Lenovo" | Allowed only filtered rows and all related transactions |

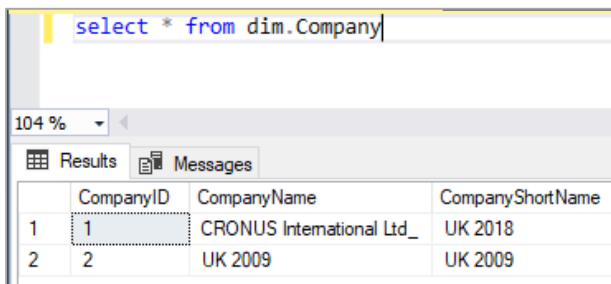
Check the **Row Filters** by clicking **top corner of DAX Filter** to **sort rows** by DAX text:



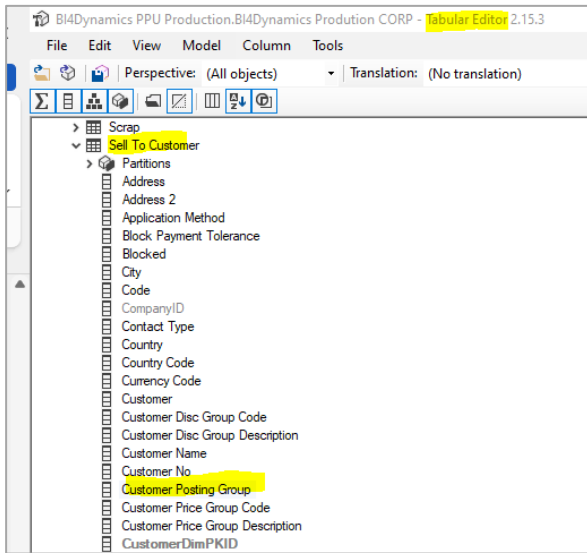
2.5 Obtain the correct name of the desired filtering column

When **filtering the rows by column in a table**, the exact column name should be used as in tabular database. This **may not be** the **column name** in front-end tool (Excel or Power BI) as this may be a **caption** (label) or **translation**.

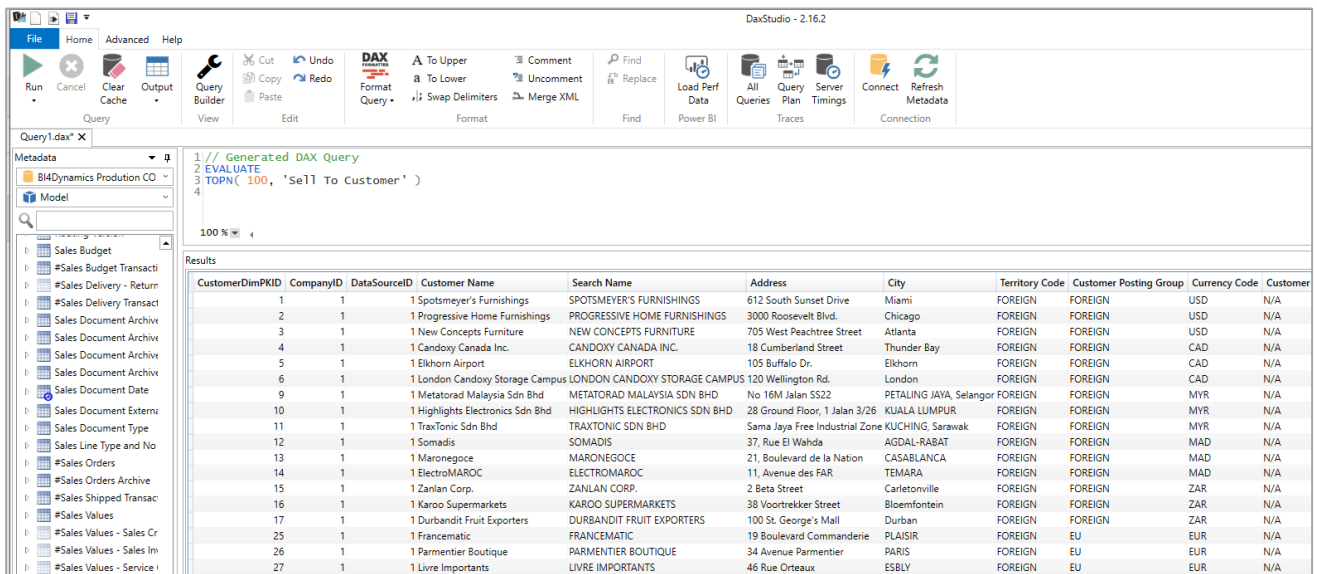
Column names from **SQL** will not work, as **names** in data warehouse **do not have spaces**.



Columns can be explored by Tabular Editor:

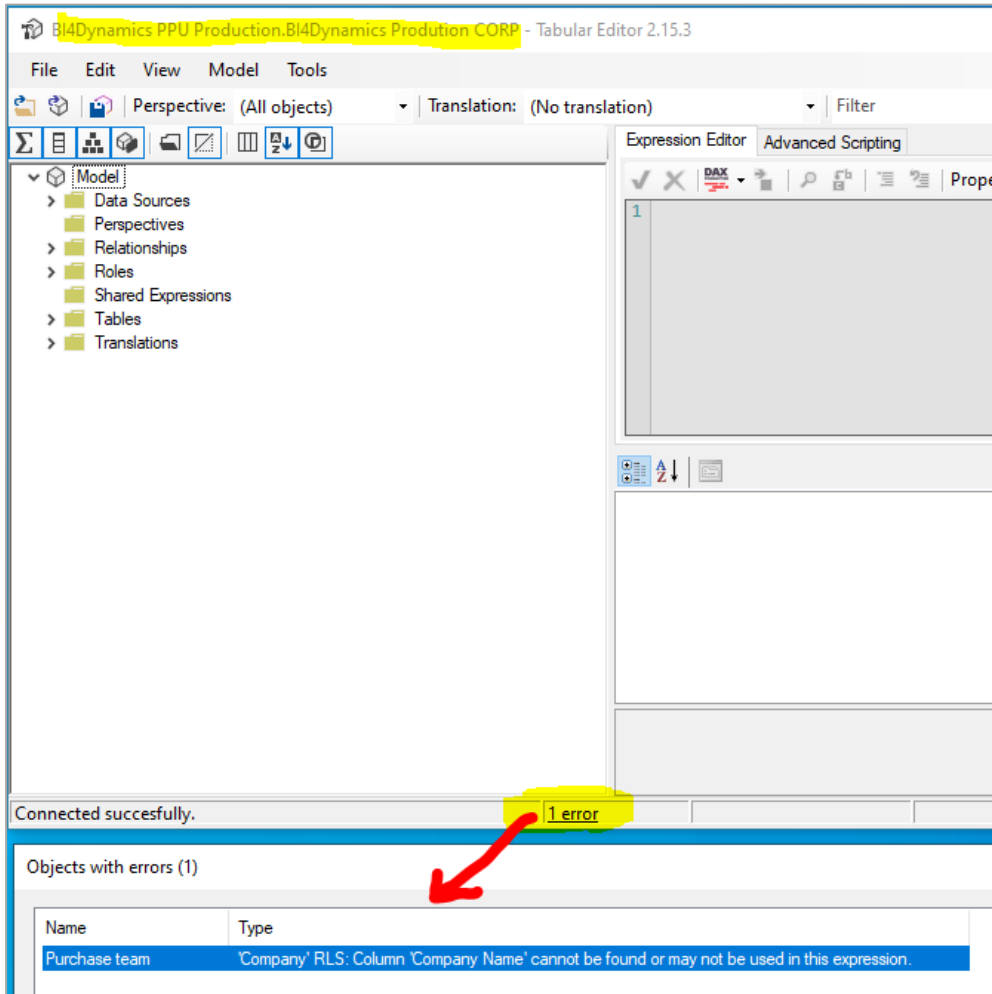


Or with DAX Studio:



2.6 Check the syntax of DAX Filters

Use **Tabular editor** to check the accuracy of the **DAX syntax**.



3 SETTING UP DYNAMIC ROLES

3.1 Identifying connected user

Dynamics roles are most often based on user login and most used identifiers are:

- A.** On-premises Analysis Services will use **Windows** login,
DAX function to identify user is **USERNAME()**
- B.** Azure Analysis Services and Power BI Service is using **Azure AD** login,
DAX function to identify user is **USERPRINCIPALNAME()**

3.2 Bridge table

Tabular database must have a table which makes the bridge between User and related permission.
Example of a table **Salesperson** where columns contain both (Windows and AAD) user mappings.

| Salesperson name | E-mail | Windows login |
|------------------|----------------|--------------------|
| John Brown | jb@contoso.com | CONTOSO\JOHNBROWN |
| Mary Winter | mw@contoso.com | CONTOSO\MARYWINTER |

3.3 DAX expression for Dynamic security

DAX Filter examples for Dynamics security (2 options):

| Table | DAX Filter expression | Login type | Tabular database installation option |
|-------------|---|------------|--------------------------------------|
| Salesperson | = [Windows login] = USERNAME() | Windows | On-premises |
| Salesperson | = [E-mail] = USERPRINCIPALNAME() | AAD | Azure or Power BI Service |

Example of Dynamics security settings with Windows login option:

The screenshot shows the 'Role Properties - Sales team - Dynamic' window in Dynamics CRM. The left sidebar has 'General', 'Membership', and 'Row Filters' options. The main area is titled 'Script' and contains a DAX filter configuration table. Below the table, there is a 'DAX Filter' field with the expression '=([Windows Login] = USERNAME())'.

| Table | DAX Filter |
|-----------------------------|---------------------------------|
| Salesperson | =([Windows Login] = USERNAME()) |
| Company | |
| FA Maintenance Registration | |
| Date | |
| Vendor | |
| Fixed Asset | |
| FA Maintenance Entry | |
| FA Maintenance | |
| Depreciation Book | |
| FA Class | |
| Reason Code | |
| Source Code | |

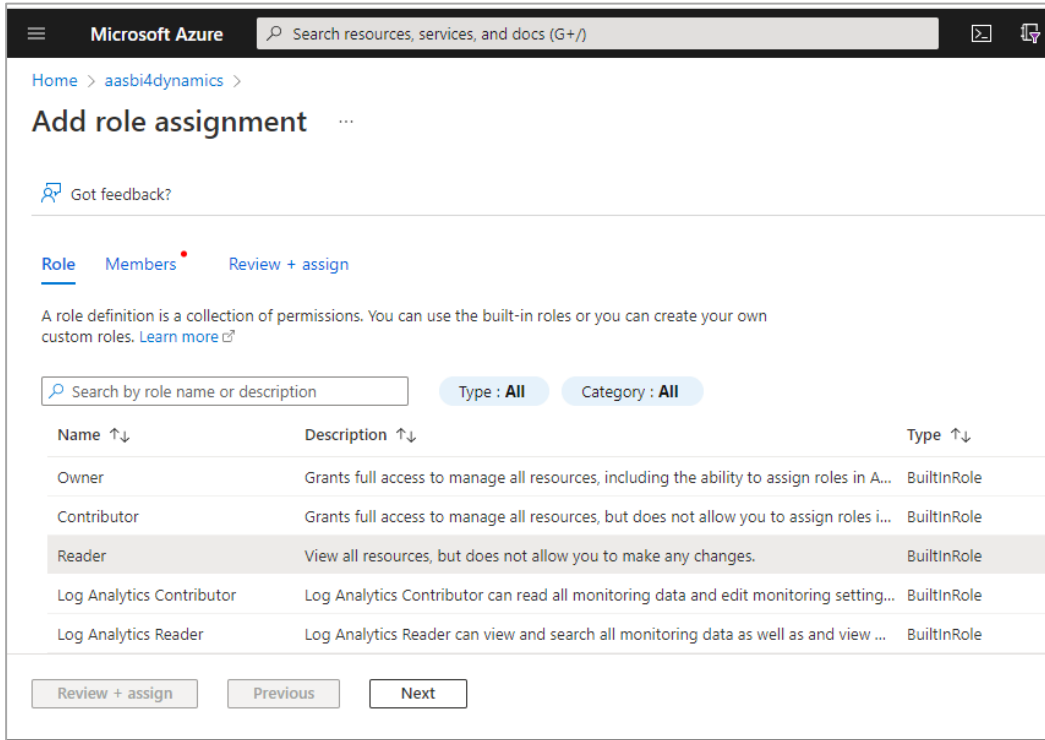
DAX Filter for Salesperson

=([Windows Login] = USERNAME())

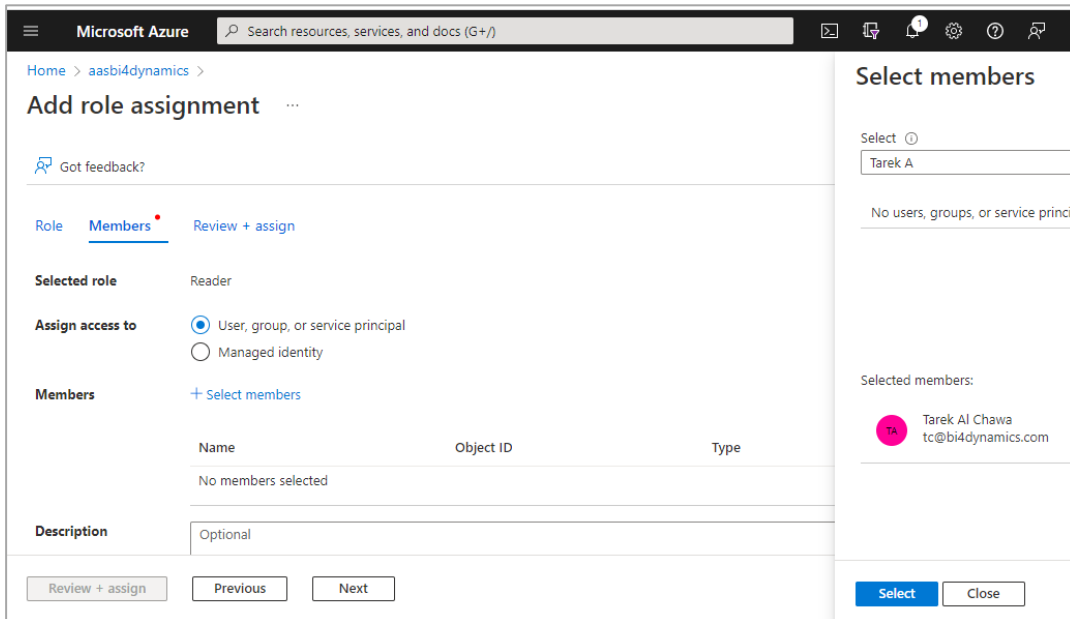
Connection details:
Server: MLADENNUC\BC
Connection: NPS-GROUP\mgvozden
View connection properties

4 OPTION B: SETTING IN AZURE PORTAL WHEN USING AZURE ANALYSIS SERVICES

When using **Azure Analysis services** users **must be added** to Azure Analysis Services in **Azure Portal**, setting **user type = Reader**.

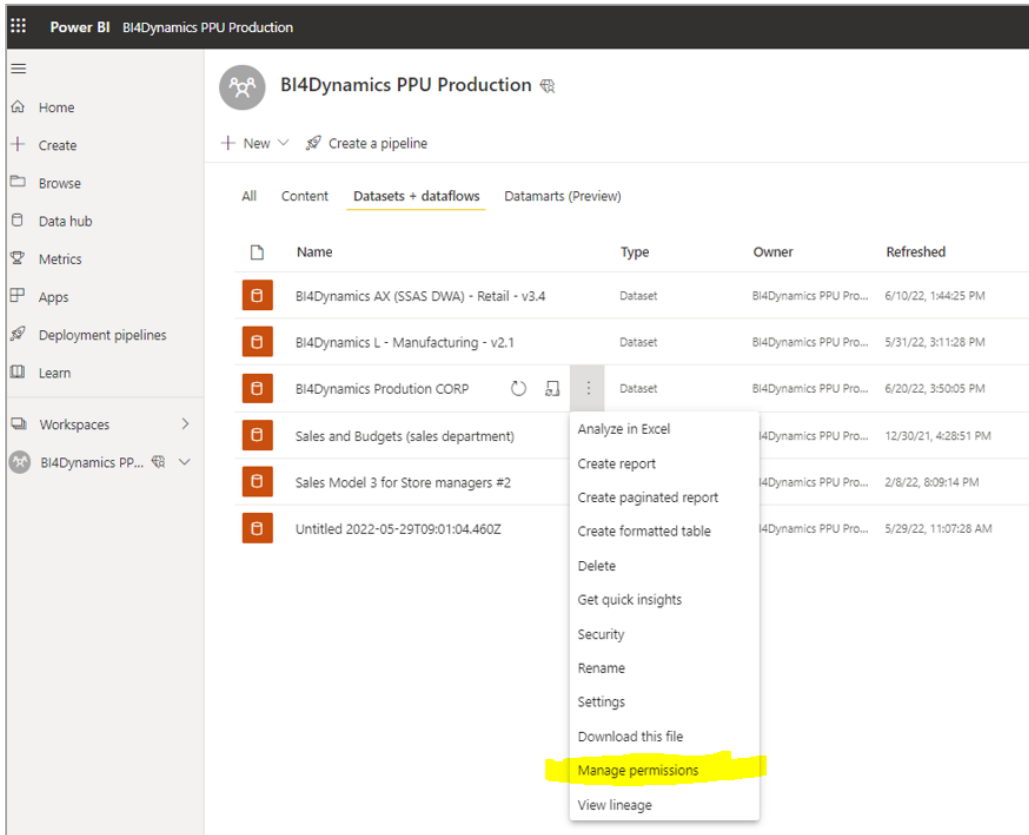


And **select the members:**

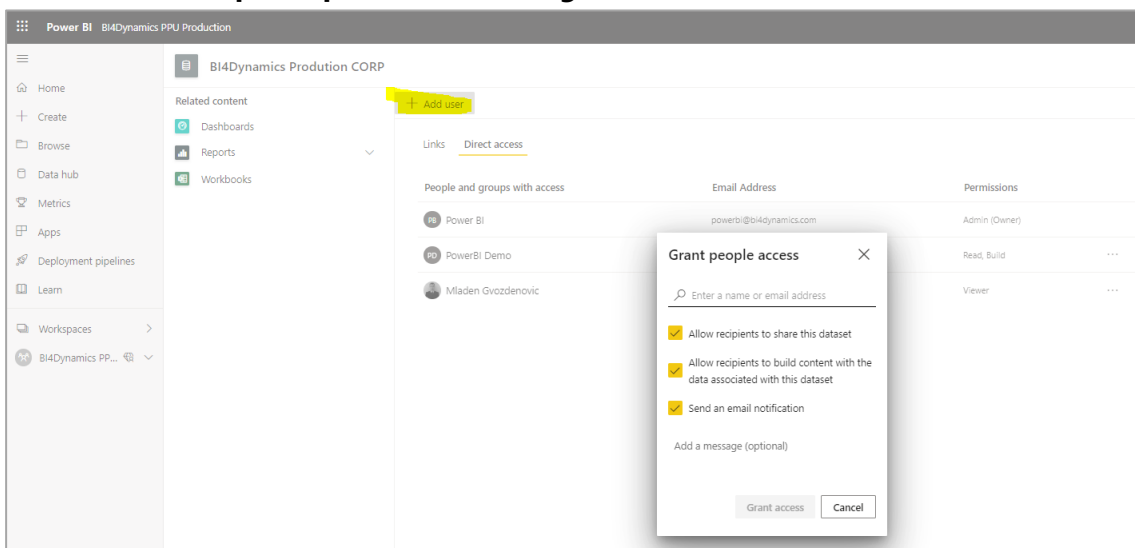


5 OPTION C: SETTING IN POWER BI SERVICE WHEN USING POWER BI PREMIUM

Go to **workspace**, select the **DataSource** and go to **Manage Permissions**:



Add user with required permission settings.



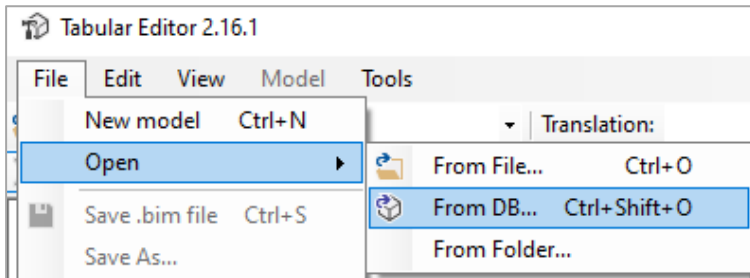
Please note that user with **Read** permission must have **Build** option selected as well.

6 SETTING UP ROLES WITH OBJECT-LEVEL SECURITY USING TABULAR EDITOR

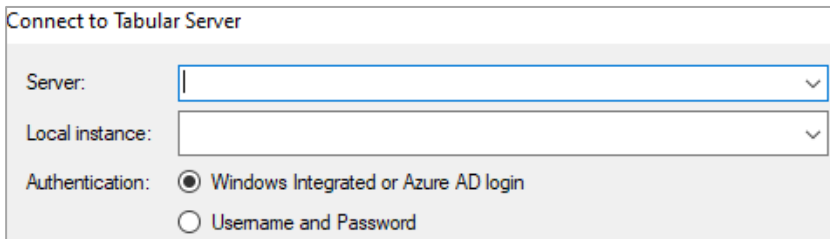
Object-level security (OLS) allows us to restrict access to specific tables and columns. Once OLS is implemented, the chosen table is secured so the existence of that object is completely hidden from the user. The users will not be able to access the table through any reporting tool.

Tabular editor can be used to implement OLS.

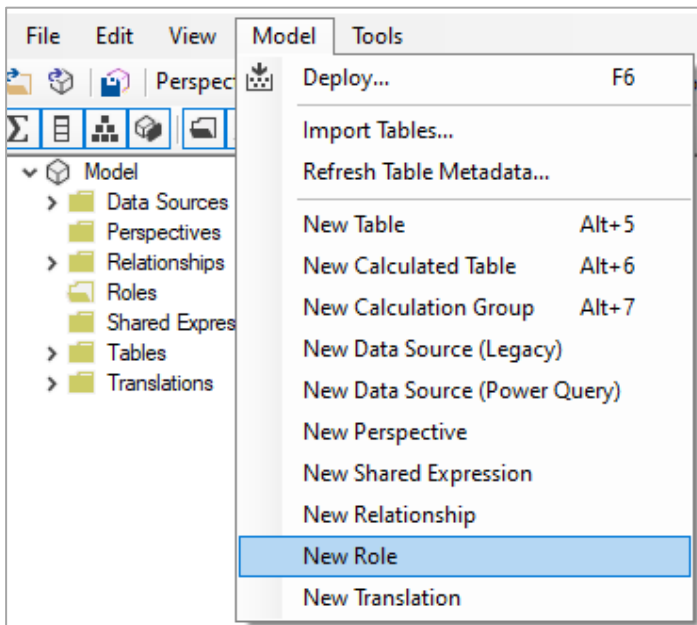
Open **Tabular editor** and go to **File > Open > From DB**



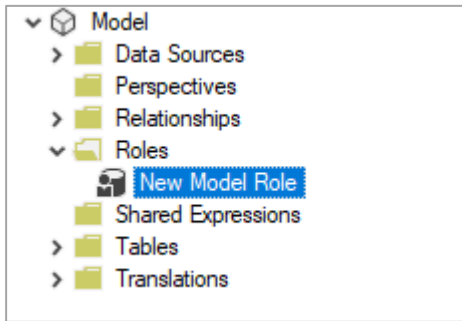
Specify the **Server** and **Instance** you want to connect to.



Select **Model** and add a **New Role**.

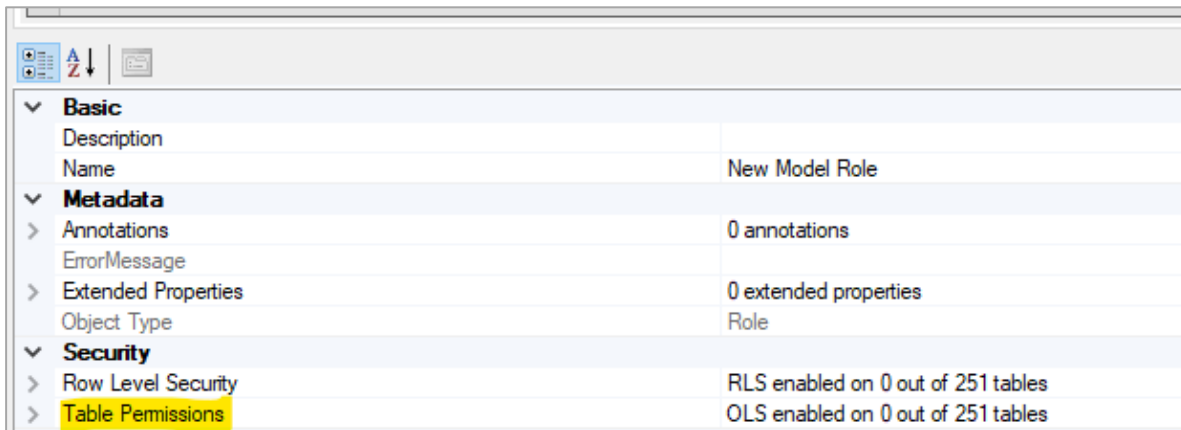


Click on a **New Model Role**.



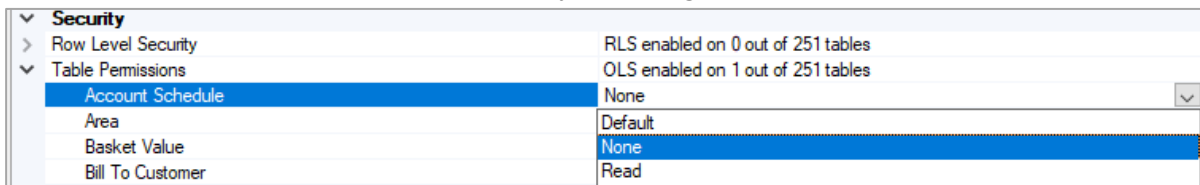
The properties of the new role will appear on the bottom left part of the application.

Under the property **Security** there will be an option **Table Permissions** available. This option should be chosen to protect objects by hiding and restricting them from users.

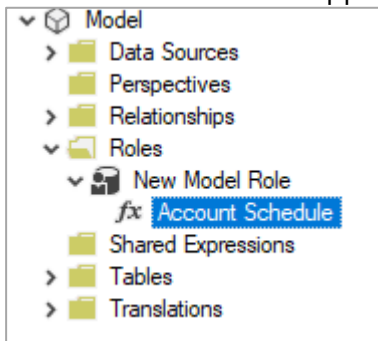


To enable object-level security, extend Table Permissions. The list of all tables in the model will appear.

The access to the tables can be restricted by choosing option **None**.

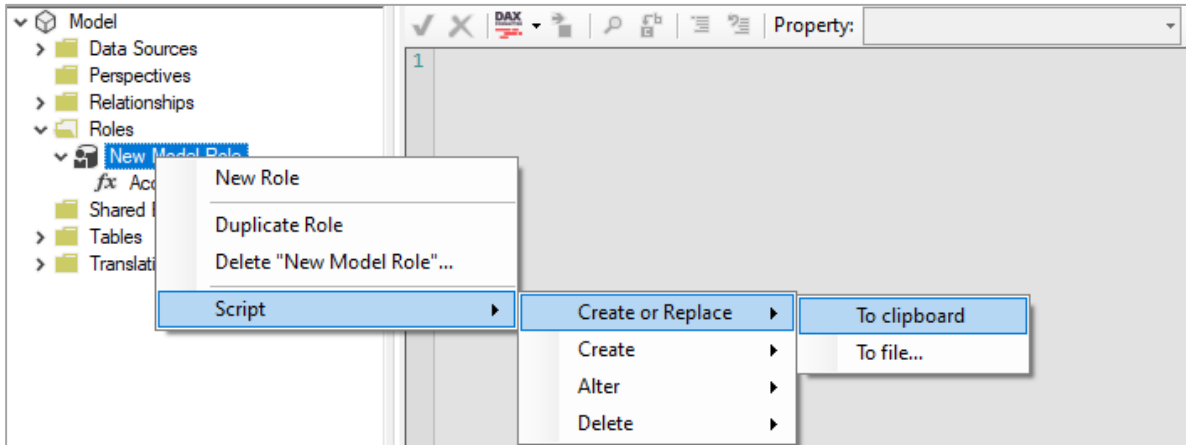


The restricted table will appear under the role.

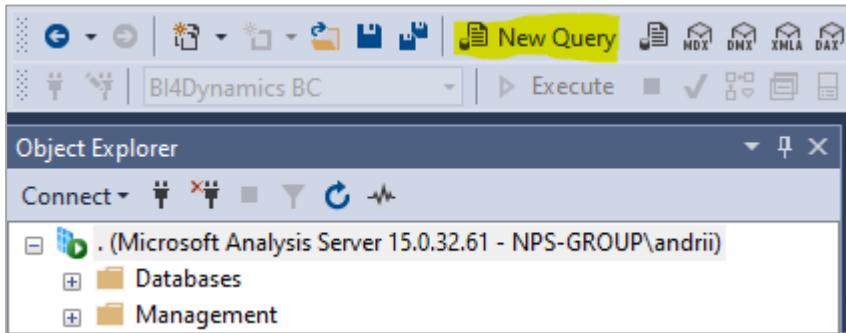


Right click the new role, select **Script > Create or Replace > To Clipboard**.

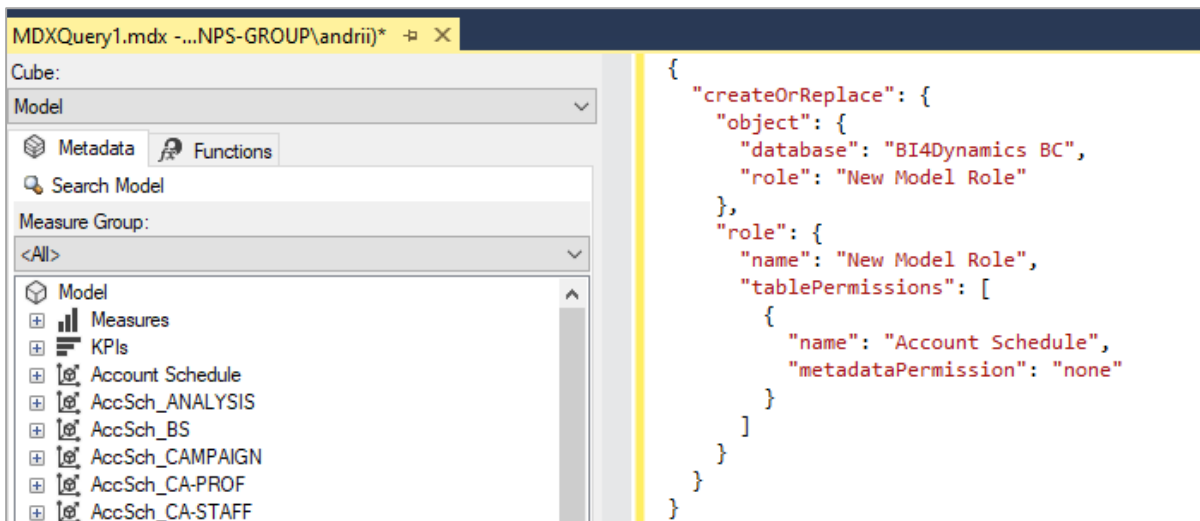
It will generate the script that creates a new role with the specified restrictions in a model.



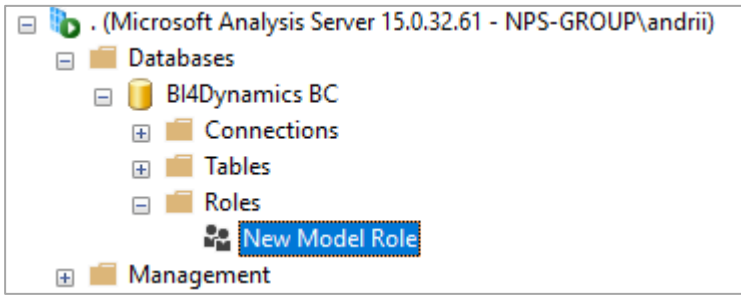
Open Microsoft SQL Server Management Studio and connect to the Server and Analysis Database you want to create a new role with Object Level Security in.



Paste the copied script and run it.



Once the execution is complete, the new role will be added to the model.



In **Role Properties** set up the access as **Read** and add members as was explained previously.

