

# Securing Content with Row Level Security



# Agenda

- User Authentication and Identity
- Power BI Tenant Administration
- Row Level Security
- Dynamic Row Level Security
- Embedding RLS-enabled Reports



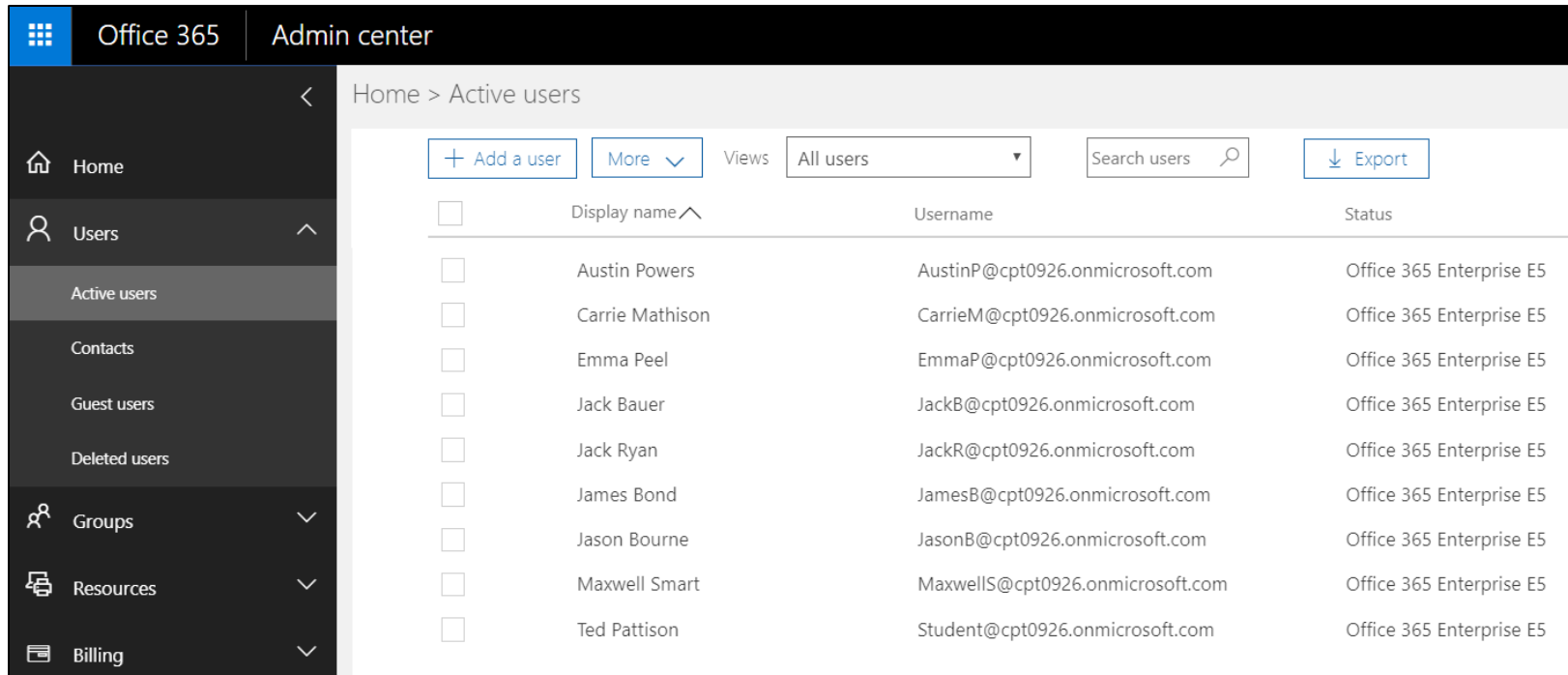
# Power BI Built on Azure Active Directory

- Azure AD manages identity in Microsoft cloud
  - Organization creates user accounts & groups in Azure AD
  - Users accounts and groups created in scope of tenant
  - Azure AD provides user authentication service
- Azure AD manages licensing and permissions
  - Provides users authorized access to Office 365
  - Provides users authorized access to SharePoint Online
  - Provides users authorized access to Dynamics 365
  - Provides users authorized access to Power BI



# Azure AD User Accounts and Licensing

- User account created within scope of tenant
  - Office 365 admin create accounts and assigns licenses



The screenshot shows the Office 365 Admin center interface. The left sidebar contains navigation links: Home, Users, Active users (selected), Contacts, Guest users, Deleted users, Groups, Resources, and Billing. The main content area is titled 'Home > Active users'. It includes a '+ Add a user' button, a 'More' dropdown, a 'Views' dropdown set to 'All users', a 'Search users' search bar, and an 'Export' button. Below these controls is a table of active users.

<input type="checkbox"/>	Display name ^	Username	Status
<input type="checkbox"/>	Austin Powers	AustinP@cpt0926.onmicrosoft.com	Office 365 Enterprise E5
<input type="checkbox"/>	Carrie Mathison	CarrieM@cpt0926.onmicrosoft.com	Office 365 Enterprise E5
<input type="checkbox"/>	Emma Peel	EmmaP@cpt0926.onmicrosoft.com	Office 365 Enterprise E5
<input type="checkbox"/>	Jack Bauer	JackB@cpt0926.onmicrosoft.com	Office 365 Enterprise E5
<input type="checkbox"/>	Jack Ryan	JackR@cpt0926.onmicrosoft.com	Office 365 Enterprise E5
<input type="checkbox"/>	James Bond	JamesB@cpt0926.onmicrosoft.com	Office 365 Enterprise E5
<input type="checkbox"/>	Jason Bourne	JasonB@cpt0926.onmicrosoft.com	Office 365 Enterprise E5
<input type="checkbox"/>	Maxwell Smart	MaxwellS@cpt0926.onmicrosoft.com	Office 365 Enterprise E5
<input type="checkbox"/>	Ted Pattison	Student@cpt0926.onmicrosoft.com	Office 365 Enterprise E5



# Multifactor Authentication


- Enabled through admin portal
  - Requires Office 365 or Azure AD Premium

## multi-factor authentication

users service settings

Note: only users licensed to use Microsoft Online Services are eligible for Multi-Factor Authentication. [Learn more about how to license other users.](#)  
Before you begin, take a look at the [multi-factor auth deployment guide](#).

[bulk update](#)


View: Sign-in allowed users  Multi-Factor Auth status: Any

<input type="checkbox"/>	DISPLAY NAME ▲	USER NAME	MULTI-FACTOR AUTH STATUS
<input type="checkbox"/>	Austin Powers	AustinP@cpt0926.onmicrosoft.com	Disabled
<input checked="" type="checkbox"/>	Carrie Mathison	CarrieM@cpt0926.onmicrosoft.com	Disabled
<input type="checkbox"/>	Emma Peel	EmmaP@cpt0926.onmicrosoft.com	Disabled
<input type="checkbox"/>	Jack Bauer	JackB@cpt0926.onmicrosoft.com	Disabled
<input type="checkbox"/>	Jack Ryan	JackR@cpt0926.onmicrosoft.com	Disabled
<input type="checkbox"/>	James Bond	JamesB@cpt0926.onmicrosoft.com	Disabled

### Carrie Mathison

CarrieM@cpt0926.onmicrosoft.com

**quick steps**

[Enable](#) 

[Manage user settings](#)

# Agenda

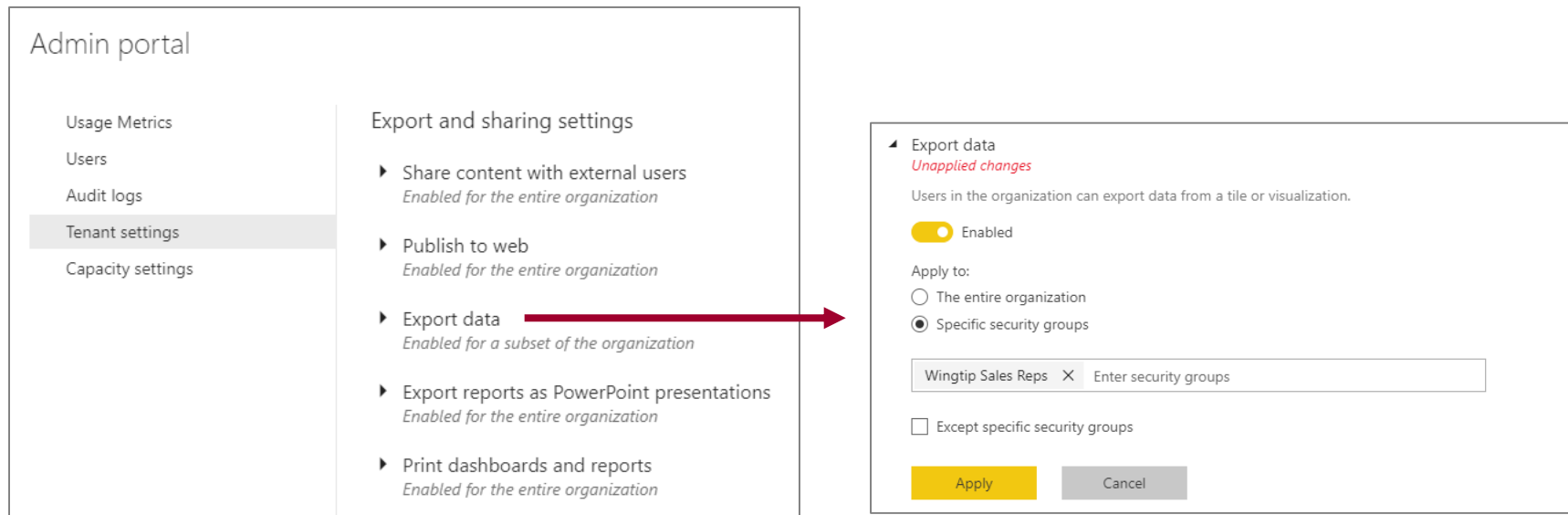
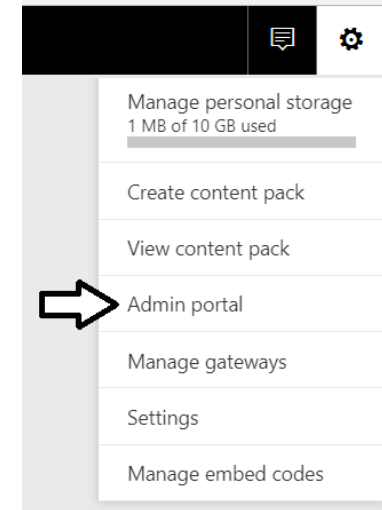
- ✓ User Authentication and Identity
- Power BI Tenant Administration
  - Row Level Security
  - Dynamic Row Level Security
  - Embedding RLS-enabled Reports





# Power BI Admin Portal

- Power BI Admins control tenant-level settings
  - Control whether users can self-register for Power BI
  - Control who can publish to web
  - Control who can export & share content
  - Control who can create content packs



# Power BI Audit Log

## Admin portal

Usage Metrics

Users

Audit logs

Tenant settings

Capacity settings

### Audit and usage settings

- ▶ Create audit logs for internal activity auditing and compliance  
*Enabled for the entire organization*
- ▶ Usage metrics for content creators  
*Enabled for the entire organization*
- ▶ Per-user data in usage metrics for content creators  
*Enabled for the entire organization*

Home > Audit log search

## Audit log search

Need to find out if a user deleted a document or if an admin reset someone's password? Search the Office 365 audit log to find out what the users and admins in your organization have done. Search for activities such as permissions, directory services, and much more. [Learn more about searching the audit log](#)

### Search

↺ Clear

Activities

Viewed Power BI report ▼

Start date

2017-10-04



00:00 ▼

End date

2017-10-12



00:00 ▼

### Results 5 results found

Date ▼	IP address	User	Activity
2017-10-04 12:00:02	10.0.0.83	<a href="#">JamesB@cpt0926.onmicrosoft.c...</a>	Viewed Power BI report
2017-10-04 11:52:42	10.0.0.83	<a href="#">JamesB@cpt0926.onmicrosoft.c...</a>	Viewed Power BI report
2017-10-04 11:52:17	10.0.0.81	<a href="#">JamesB@cpt0926.onmicrosoft.c...</a>	Viewed Power BI report
2017-10-04 10:32:03	10.0.0.79	<a href="#">JamesB@cpt0926.onmicrosoft.c...</a>	Viewed Power BI report





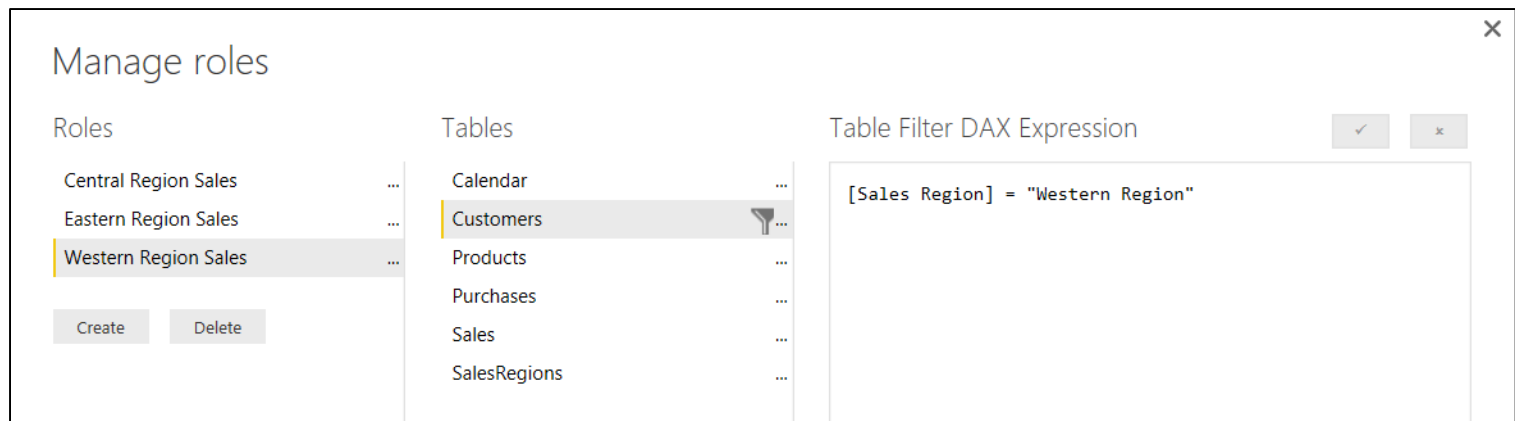
# Agenda

- ✓ User Authentication and Identity
- ✓ Power BI Tenant Administration
- Row Level Security
  - Dynamic Row Level Security
  - Embedding RLS-enabled Reports

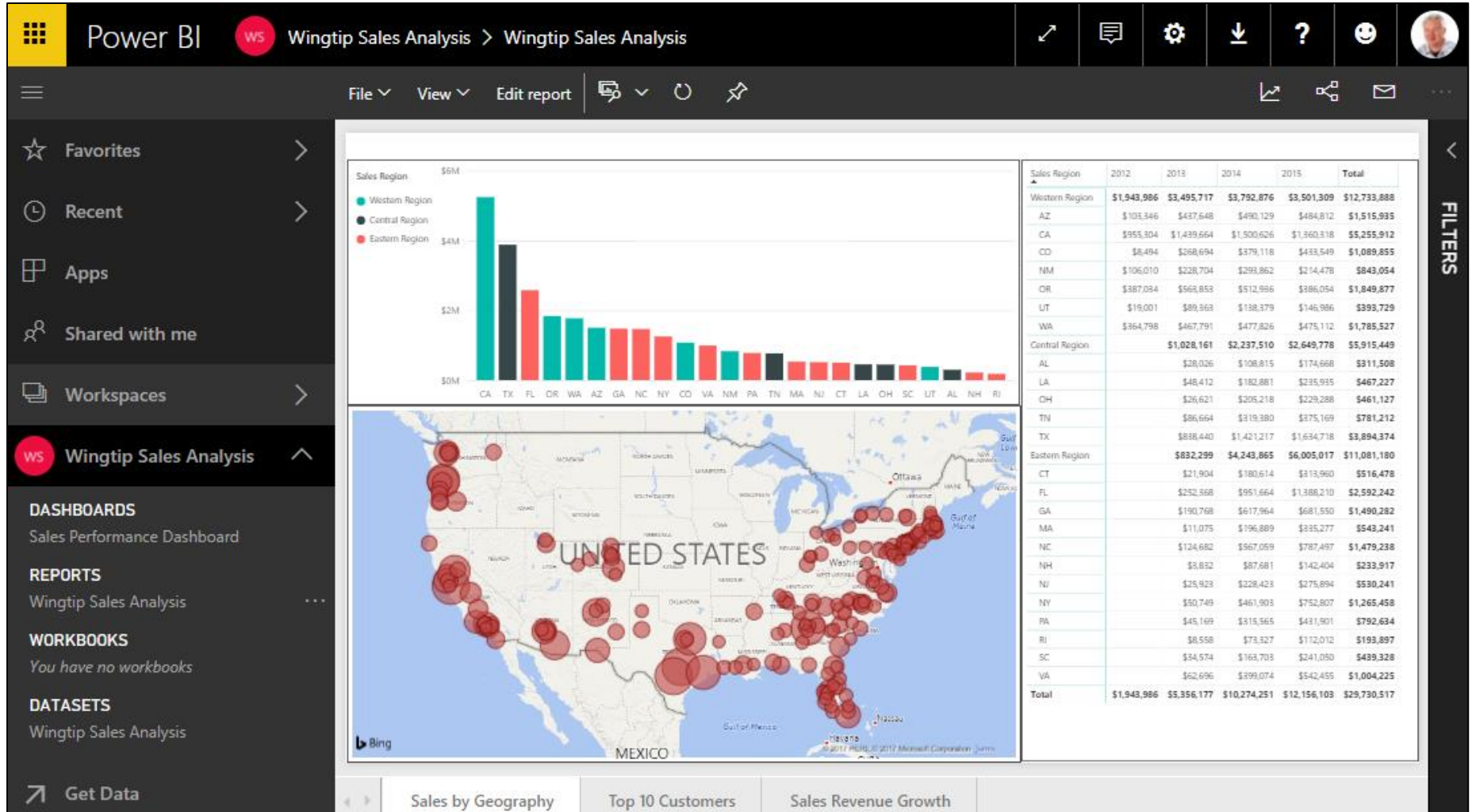


# What Is Row-level Security (RLS)

- Security Scheme based on Named Roles
  - Roles are defined using Power BI Desktop
  - Each role is scoped to the dataset within a PBIX project
- Role defined using one or more DAX expressions
  - DAX expressions restrict which rows are accessible



# Common RLS Scenario



# Configuring RLS in the Power BI Desktop

## Manage roles

Roles

All Sales Regions

Central Sales Region

Eastern Sales Region

Western Sales Region

Create

Delete

Tables

Calendar

Customers

Products

Purchases

Sales

SalesRegions

Table filter DAX expression

[Sales Region] = "Western Region"

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



# Configuring RLS in the Power BI Service

W

Western Sales Reps  
Security group

↺

✕

[Change](#) [Delete group](#)

Name	Western Sales Reps	<a href="#">Edit</a>
Description		
Owners (2)	Maxwell Smart Ted Pattison	<a href="#">Edit</a>
Members (2)	Jack Ryan Jason Bourne	<a href="#">Edit</a>

Close

Power BI

WS Wingtip Sales Analysis > Row-Level Security

≡

☆ Favorites >

🕒 Recent >

🗃 Apps

👤 Shared with me

📁 Workspaces >

WS Wingtip Sales An... ^

Row-Level Security

All Sales Regions (0)

Central Sales Region (1)

Eastern Sales Region (1)

Western Sales Region (1)

Members (1)

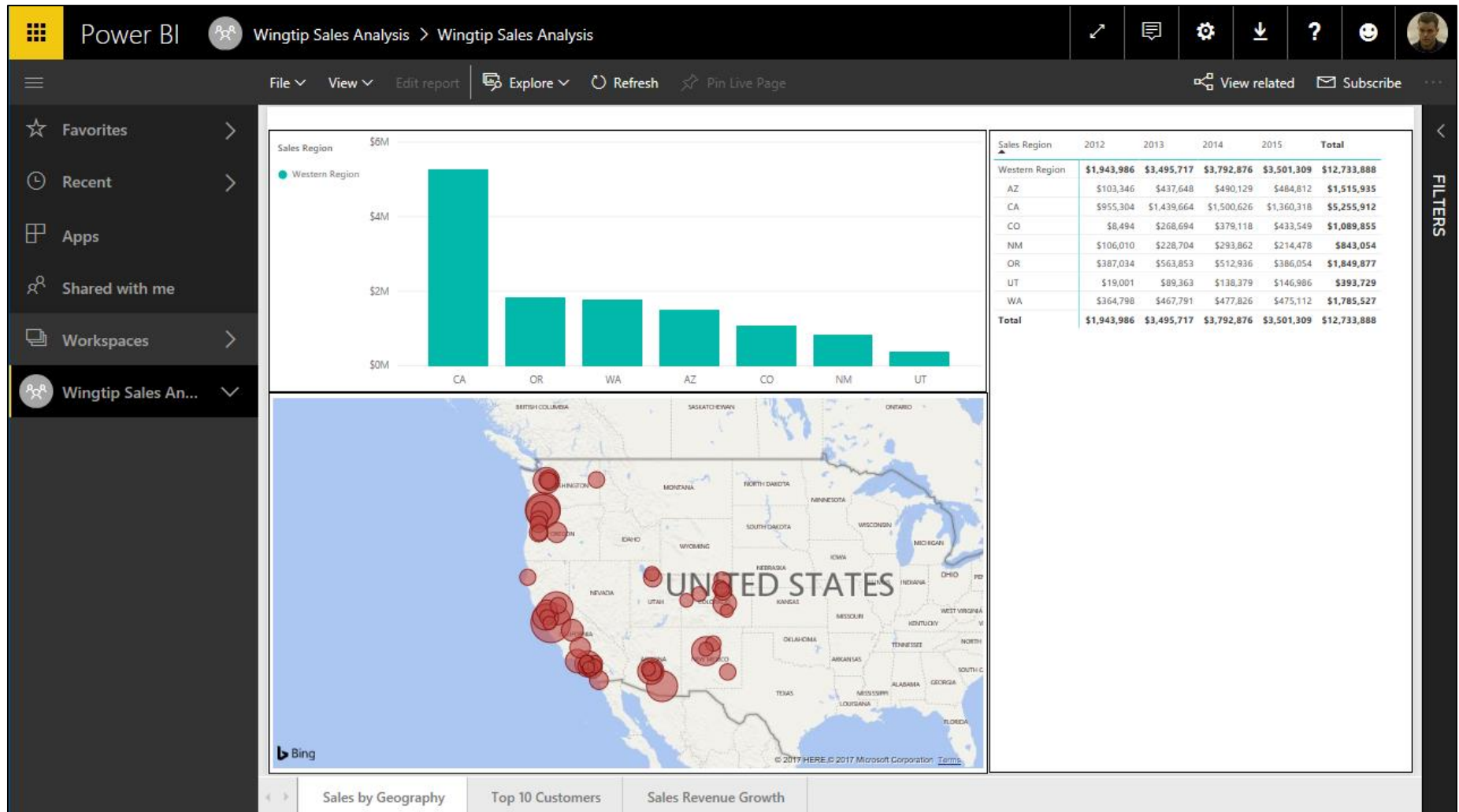
People or groups who belong to this role

Add

Western Sales Reps ✕



# RLS Enforcement







**DEMO**

## **Configuring Row-level Security**



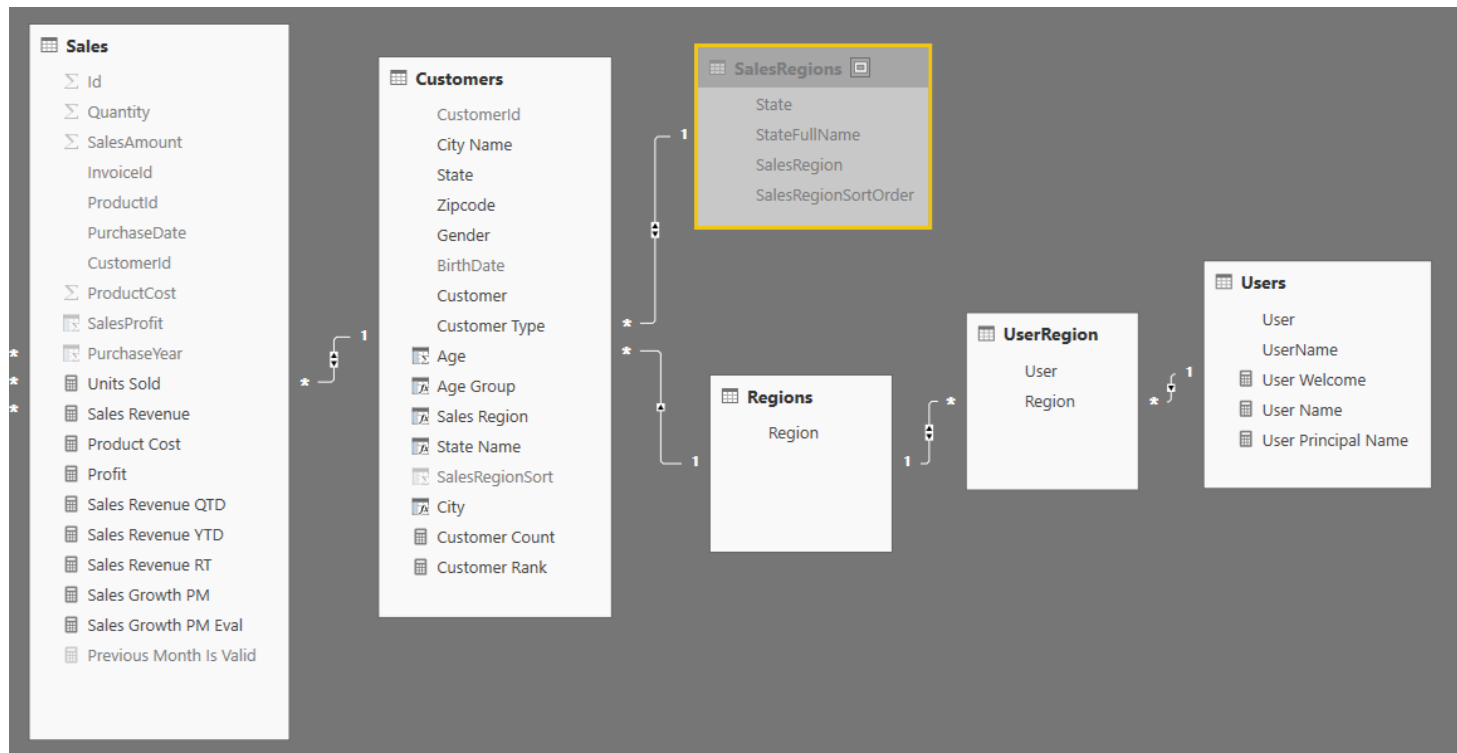
# Agenda

- ✓ User Authentication and Identity
- ✓ Power BI Tenant Administration
- ✓ Row Level Security
- Dynamic Row Level Security
- Embedding RLS-enabled Reports



# Dynamic RLS

- Design pattern for data-driven security
  - RLS set up to use login name of current user
  - Permission assignments are included as part of dataset
  - Implemented using bi-directional cross-filtering



# Configuring Cross-direction Filtering

The diagram illustrates a database schema with the following tables and attributes:

- Customers**: BirthDate, Customer, Customer Type, Age, Age Group, Sales Region, State Name, SalesRegionSort, City, Customer Count, Customer Rank.
- Regions**: Region.
- UserRegion**: User, Region.
- Users**: User, UserName, User Welcome, User Name, User Principal Name.

Relationships are shown with cardinalities: Customers (1) to Regions (\*), Regions (1) to UserRegion (\*), and UserRegion (1) to Users (\*).

A red arrow points from the UserRegion relationship to the configuration dialog box below.

**UserRegion Configuration Dialog**

Relationship: UserRegion

User	Region
EmmaP@cpt0926.onmicrosoft.com	Central Region
JackB@cpt0926.onmicrosoft.com	Central Region
MaxwellS@cpt0926.onmicrosoft.com	Central Region

Regions:

Region
Western Region
Central Region
Eastern Region

Cardinality: Many to one (\*:1)

Cross filter direction: Both

☒ Make this relationship active

☐ Assume referential integrity

☒ Apply security filter in both directions

OK Cancel

# Dynamically Tracking the Current User

## Manage roles

### Roles

Dynamic RLS Role ...

CreateDelete

### Tables

Calendar ...

Customers ...

Products ...

Purchases ...

Regions ...

Sales ...

SalesRegions ...

UserRegion ...

Users ...

### Table filter DAX expression

✓✕

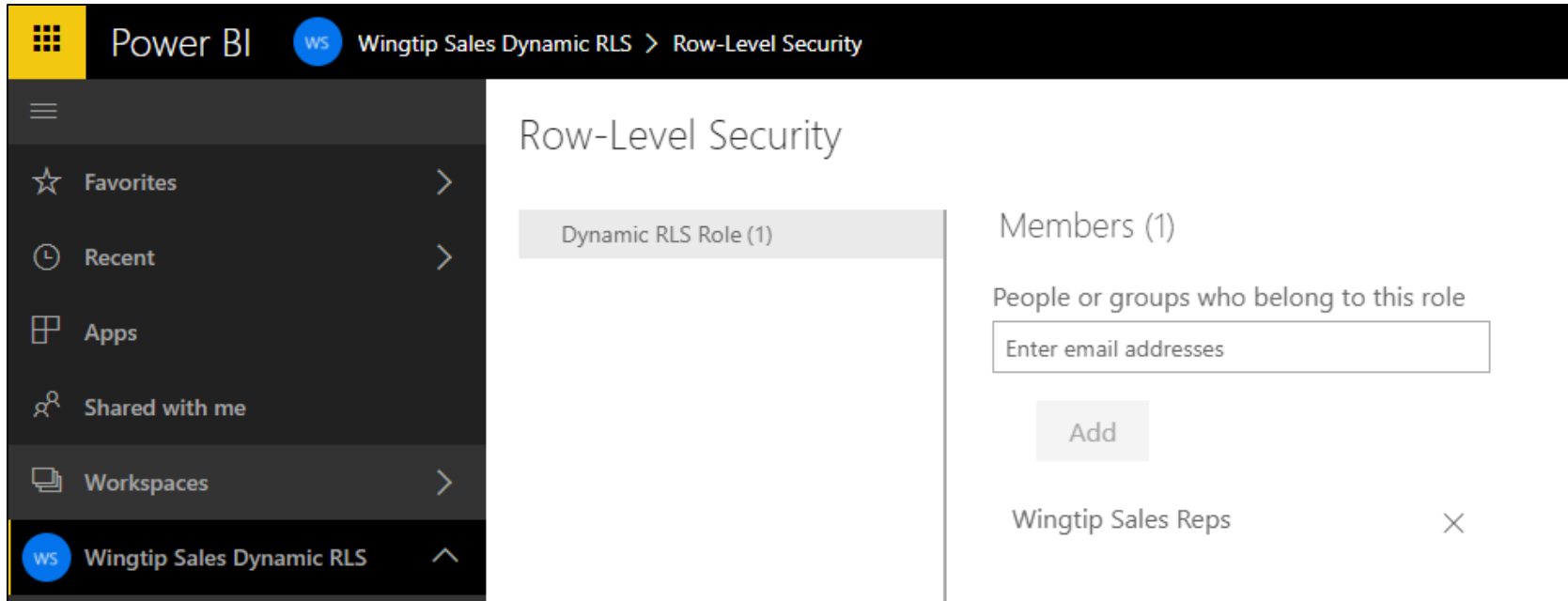
[User]=Username()

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"

SaveCancel



# All Users Must Be Added To a Role



The screenshot shows the Power BI interface for configuring Row-Level Security (RLS). The top navigation bar includes the Power BI logo, a workspace icon labeled 'ws', and the breadcrumb 'Wingtip Sales Dynamic RLS > Row-Level Security'. The left sidebar contains navigation options: Favorites, Recent, Apps, Shared with me, Workspaces, and the active workspace 'Wingtip Sales Dynamic RLS'. The main content area is titled 'Row-Level Security' and is divided into two panels. The left panel, 'Dynamic RLS Role (1)', is currently empty. The right panel, 'Members (1)', shows the list of users assigned to the role. It includes a text input field labeled 'Enter email addresses' and an 'Add' button. Below the input field, the user 'Wingtip Sales Reps' is listed with a close icon (X) to its right.

Power BI ws Wingtip Sales Dynamic RLS > Row-Level Security

Row-Level Security

Dynamic RLS Role (1)

Members (1)

People or groups who belong to this role

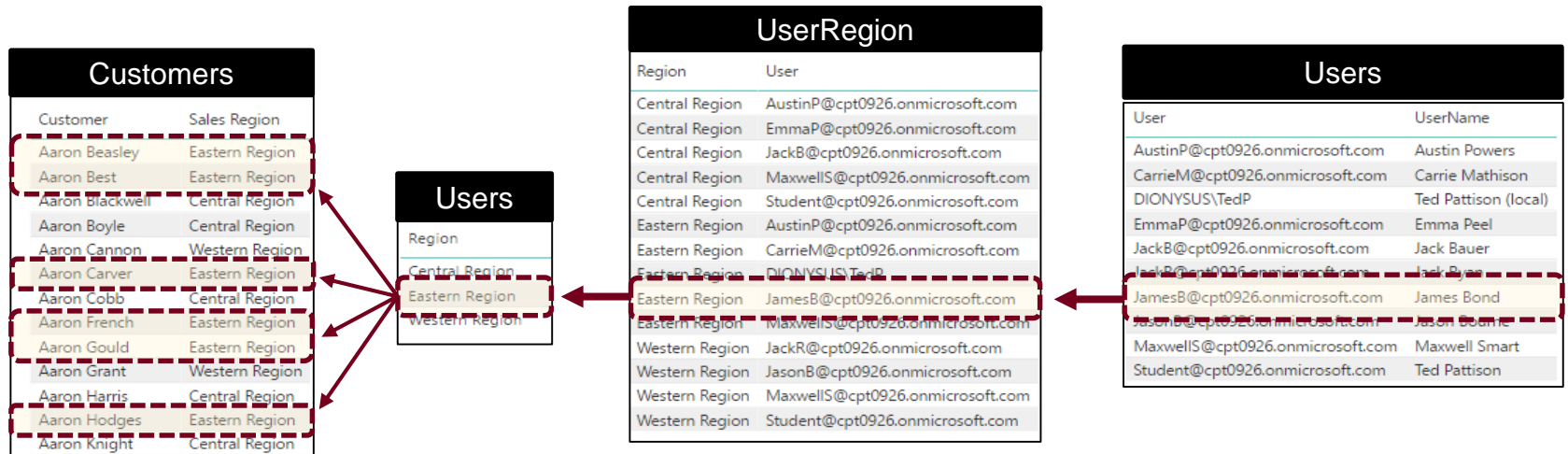
Enter email addresses

Add

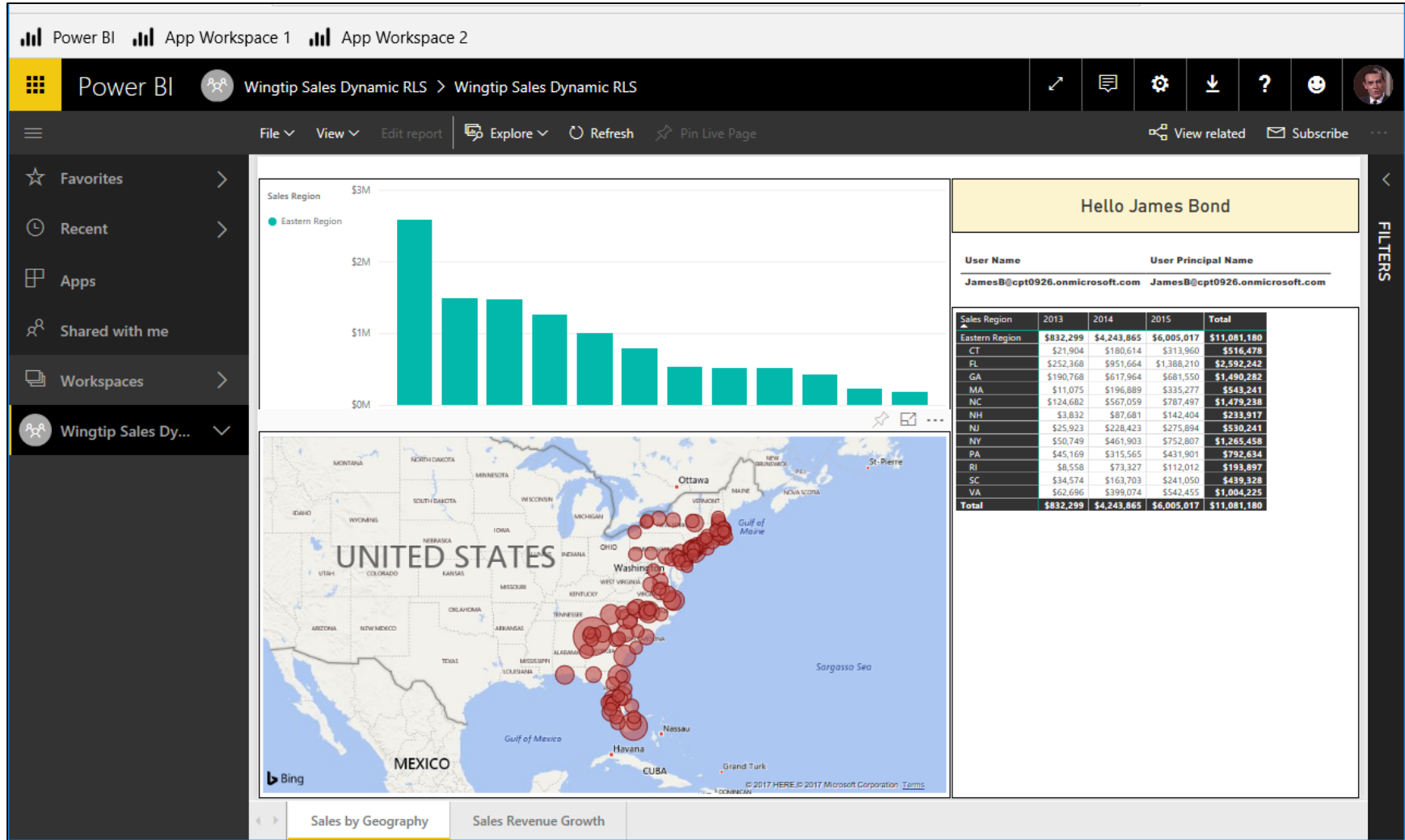
Wingtip Sales Reps X



# Dynamic RLS Table Filtering



# Dynamic RLS Enforcement







**DEMO**

# Configuring Dynamic Row-level Security

# Agenda

- ✓ User Authentication and Identity
- ✓ Power BI Tenant Administration
- ✓ Row Level Security
- ✓ Dynamic Row Level Security
- Embedding RLS-enabled Reports



# Embedding RLS-enabled Reports

- Programming with EffectiveIdentity
  - RLS-enabled reports require role(s) for access
  - Embed tokens must be generated with 1 or more roles
  - Roles added to embed token using EffectiveIdentity

```
GenerateTokenRequest generateTokenRequestParameters =  
    new GenerateTokenRequest(accessLevel: accessLevel,  
        identities: new List<EffectiveIdentity> {  
            new EffectiveIdentity(username: currentUser.UserName,  
                datasets: new List<string> { datasetId },  
                roles: roles) });  
  
string embedToken =  
    (await pbiClient.Reports.GenerateTokenInGroupAsync(workspaceId,  
        report.Id,  
        generateTokenRequestParameters)).Token;
```





# Programming with EffectiveIdentity

```
public static async Task<ReportEmbeddingData> GetReportEmbeddingDataWithRlsRoles() {  
    string currentUserName = HttpContext.Current.User.Identity.GetUserName();  
    ApplicationDbContext context = new ApplicationDbContext();  
    var userManager = new UserManager<ApplicationUser>(new UserStore<ApplicationUser>(context));  
    ApplicationUser currentUser = userManager.FindByName(currentUserName);  
  
    var roleManager = new RoleManager<IdentityRole>(new RoleStore<IdentityRole>(context));  
  
    List<string> roles = new List<string>();  
  
    foreach (var role in currentUser.Roles) {  
        roles.Add(roleManager.FindById(role.RoleId).Name);  
    }  
  
    string accessLevel = HttpContext.Current.User.IsInRole("Admin") ? "edit" : "view";  
  
    PowerBIClient pbiClient = GetPowerBiClient();  
  
    var report = await pbiClient.Reports.GetReportInGroupAsync(workspaceId, reportId);  
    var embedUrl = report.EmbedUrl;  
    var reportName = report.Name;  
    var datasetId = report.DatasetId;  
  
    GenerateTokenRequest generateTokenRequestParameters =  
        new GenerateTokenRequest(accessLevel: accessLevel,  
                                identities: new List<EffectiveIdentity> {  
                                    new EffectiveIdentity(username: currentUser.UserName,  
                                                         datasets: new List<string> { datasetId },  
                                                         roles: roles) });  
  
    string embedToken =  
        (await pbiClient.Reports.GenerateTokenInGroupAsync(workspaceId,  
                                                         report.Id,  
                                                         generateTokenRequestParameters)).Token;  
  
    return new ReportEmbeddingData {  
        reportId = reportId,  
        reportName = reportName,  
        embedUrl = embedUrl,  
        accessToken = embedToken  
    };  
}
```



# Summary

- ✓ User Authentication and Identity
- ✓ Power BI Tenant Administration
- ✓ Row Level Security
- ✓ Dynamic Row Level Security
- ✓ Embedding RLS-enabled Reports

