

# Securing Power BI Embedded Solutions using Row Level Security



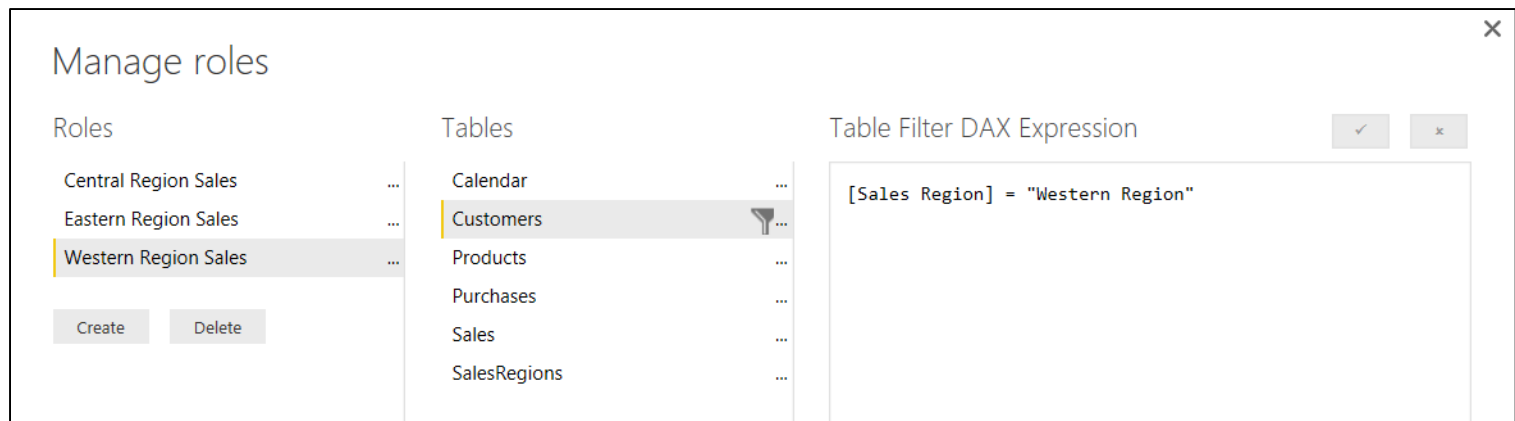
# Agenda

- Understanding Row-Level Security (RLS) in Power BI
- Adding RLS Roles to a Power BI Desktop Project
- Designing a Dynamic RLS Security Scheme
- RLS with the User-Owns-Data Model
- RLS with the App-Owns-Data Model
- Generating Embed Tokens with EffectiveIdentity



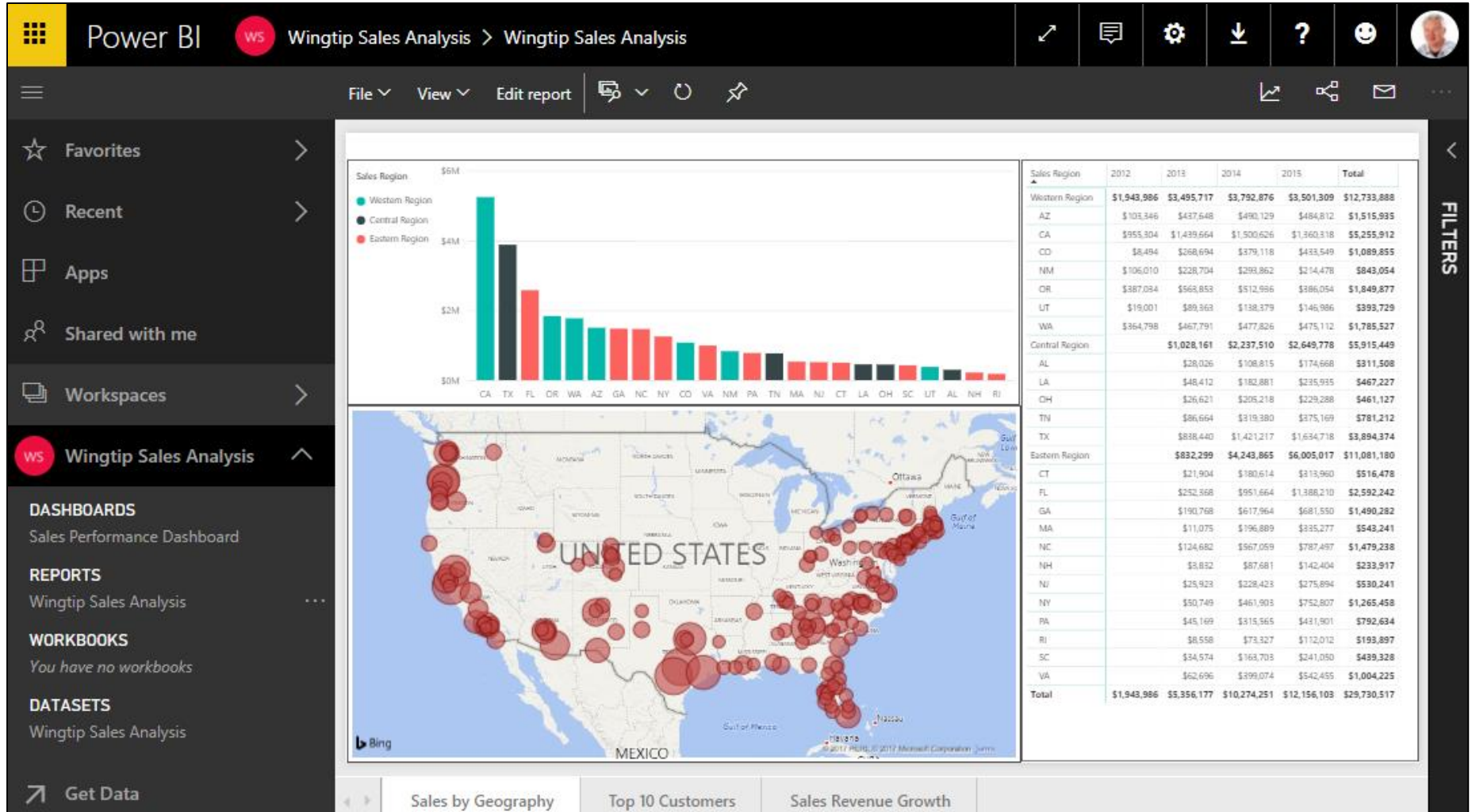
# What Is Row-level Security (RLS)

- 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<br>Ted Pattison | <a href="#">Edit</a> |
| Members (2) | Jack Ryan<br>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)

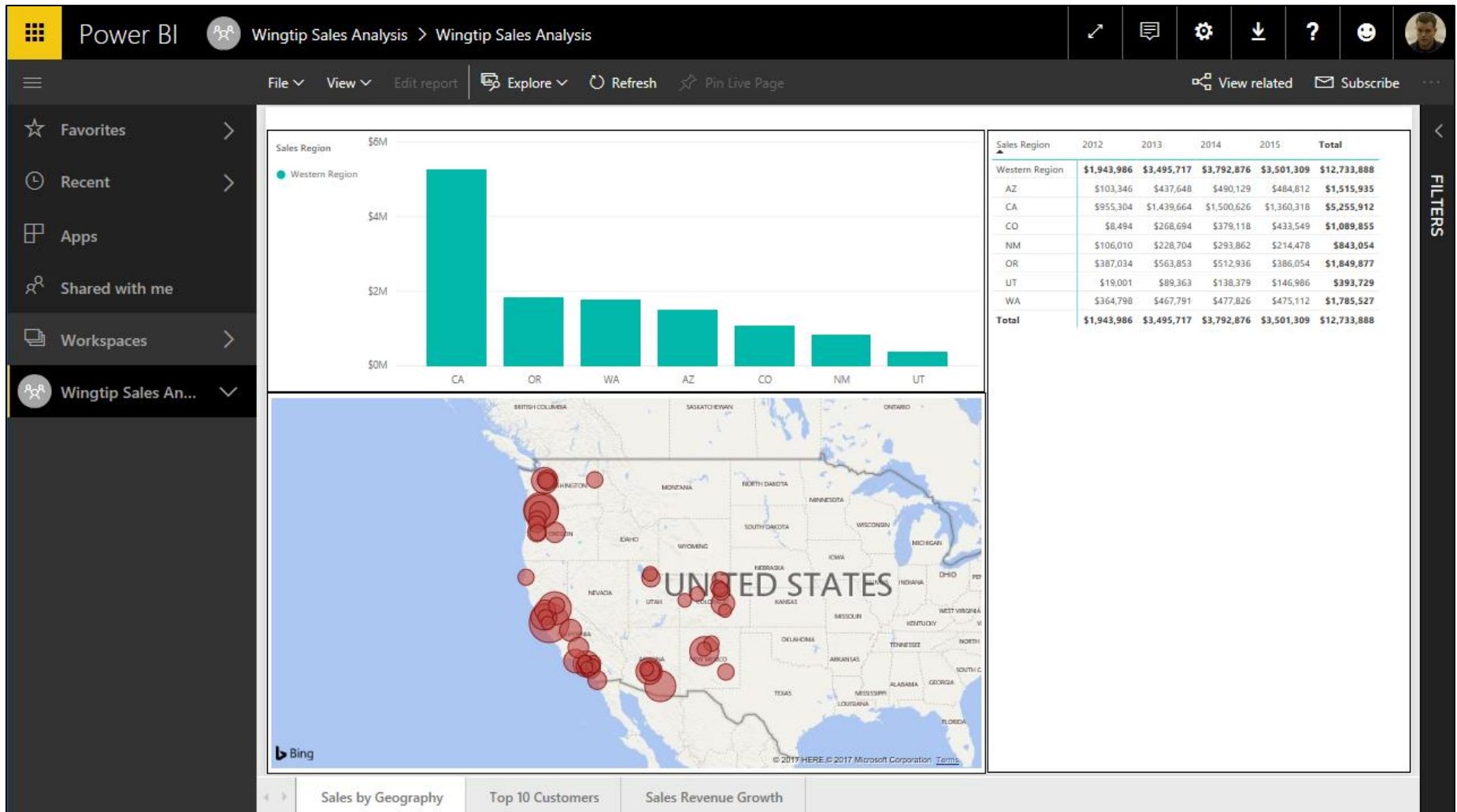
People or groups who belong to this role

Add

Western Sales Reps ✕



# RLS Enforcement







**DEMO**

## **Configuring Row-level Security**



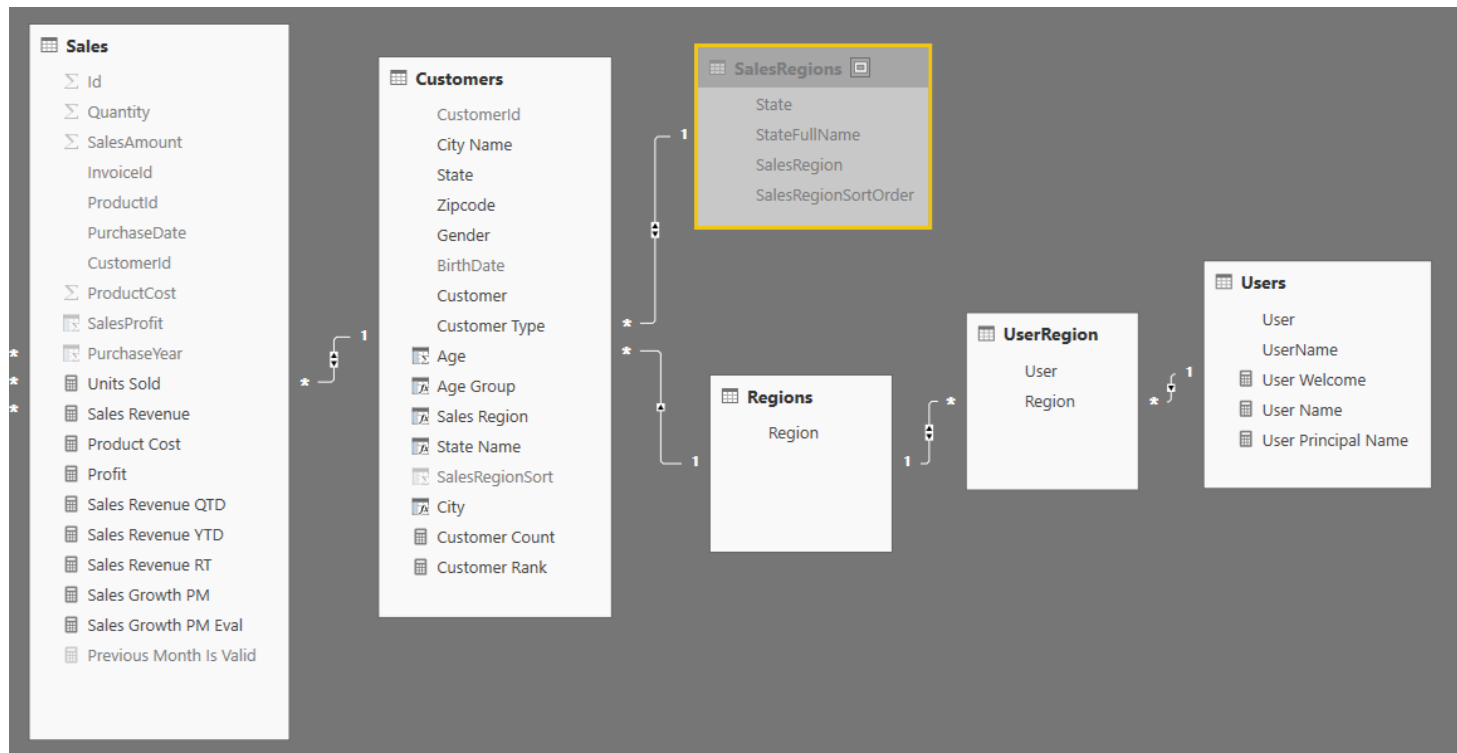
# Agenda

- ✓ User Authentication and Identity
- ✓ Power BI Tenant Administration
- ✓ Data Security
- ✓ App Workspaces
- ✓ Row Level Security
- Dynamic Row Level Security



# 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 the database schema and the configuration for cross-direction filtering on the UserRegion relationship.

**Database Schema:**

- Customers** (1) is related to **Regions** (\*) via BirthDate, Customer, and Customer Type.
- Regions** (1) is related to **UserRegion** (\*) via Region.
- UserRegion** (\*) is related to **Users** (1) via User and Region.

**UserRegion Relationship Configuration:**

**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

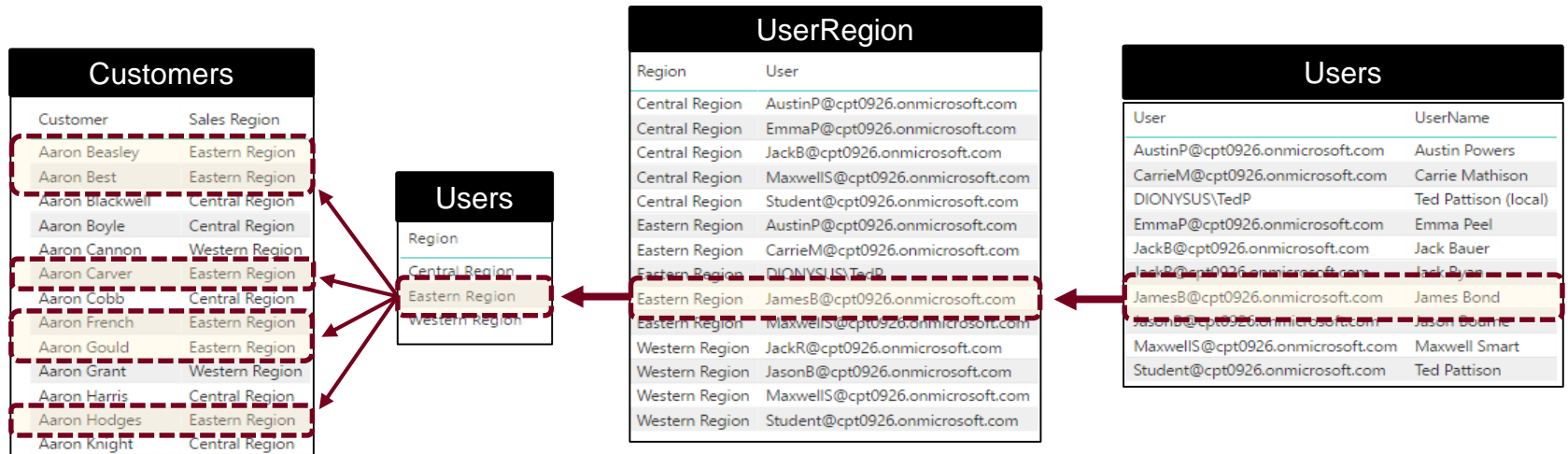
Enter email addresses

Add

Wingtip Sales Reps X

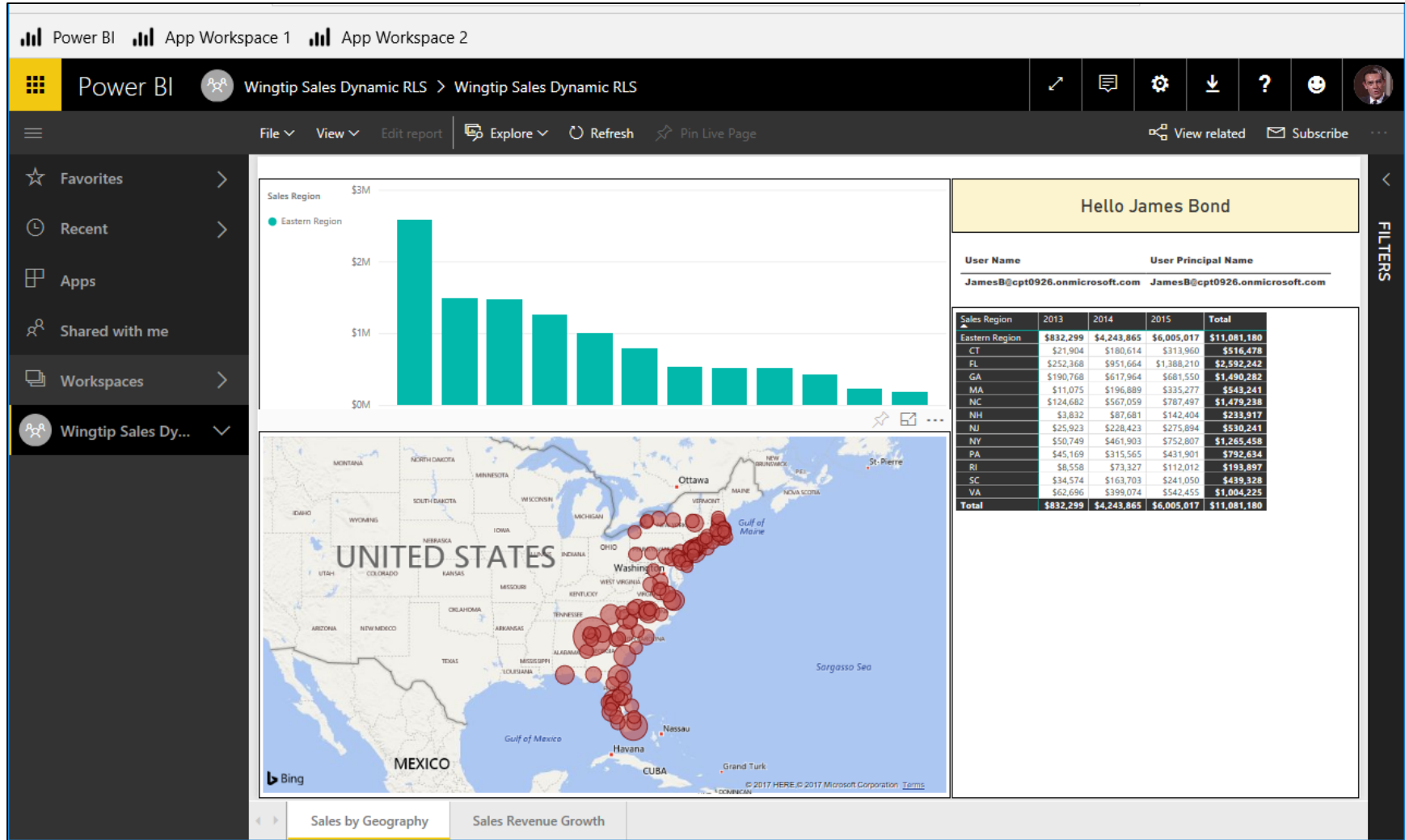


# Dynamic RLS Table Filtering





# Dynamic RLS Enforcement





**DEMO**

# Configuring Dynamic Row-level Security

# Summary

- Understanding Row-Level Security (RLS) in Power BI
- Adding RLS Roles to a Power BI Desktop Project
- Designing a Dynamic RLS Security Scheme
- RLS with the User-Owns-Data Model
- RLS with the App-Owns-Data Model
- Generating Embed Tokens with EffectiveIdentity

