Hands-on Lab: Views in PostgreSQL

Estimated time needed: 15 minutes

In this lab, you will learn how to create, execute, and materialize views in the PostgreSQL database service using the pgAdmin graphical user interface (GUI) tool. Materialized views behave differently compared to regular views. The result set is materialized or saved for future use in the materialized views. You can not insert, update, or delete rows like in regular views. Materialized views store the results of a database query as a separate table-like object so that someone can access the results later without having to re-run the query. As a result, materialized views can improve database performance compared to regular views.

Software used in this lab

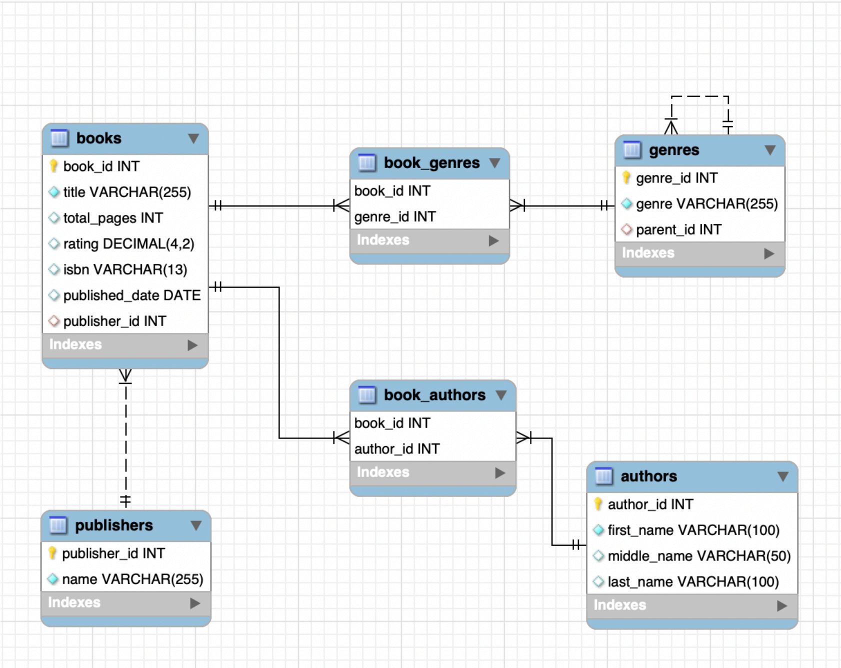
In this lab, you will use the PostgreSQL Database. PostgreSQL is a relational database management system (RDBMS) designed to store, manipulate, and retrieve data efficiently.

To complete this lab, you will utilize the PostgreSQL relational database service available as part of IBM Skills Network Labs (SN Labs) Cloud IDE. SN Labs is a virtual lab environment used in this course.

Database used in this lab

You will use the eBooks database in the lab.

The following ERD diagram shows the schema of the complete eBooks database used in this lab:



Objectives

After completing this lab, you will be able to use pgAdmin with PostgreSQL to:

- Restore a database schema and data
- Create and execute a view
- Create and execute a materialized view

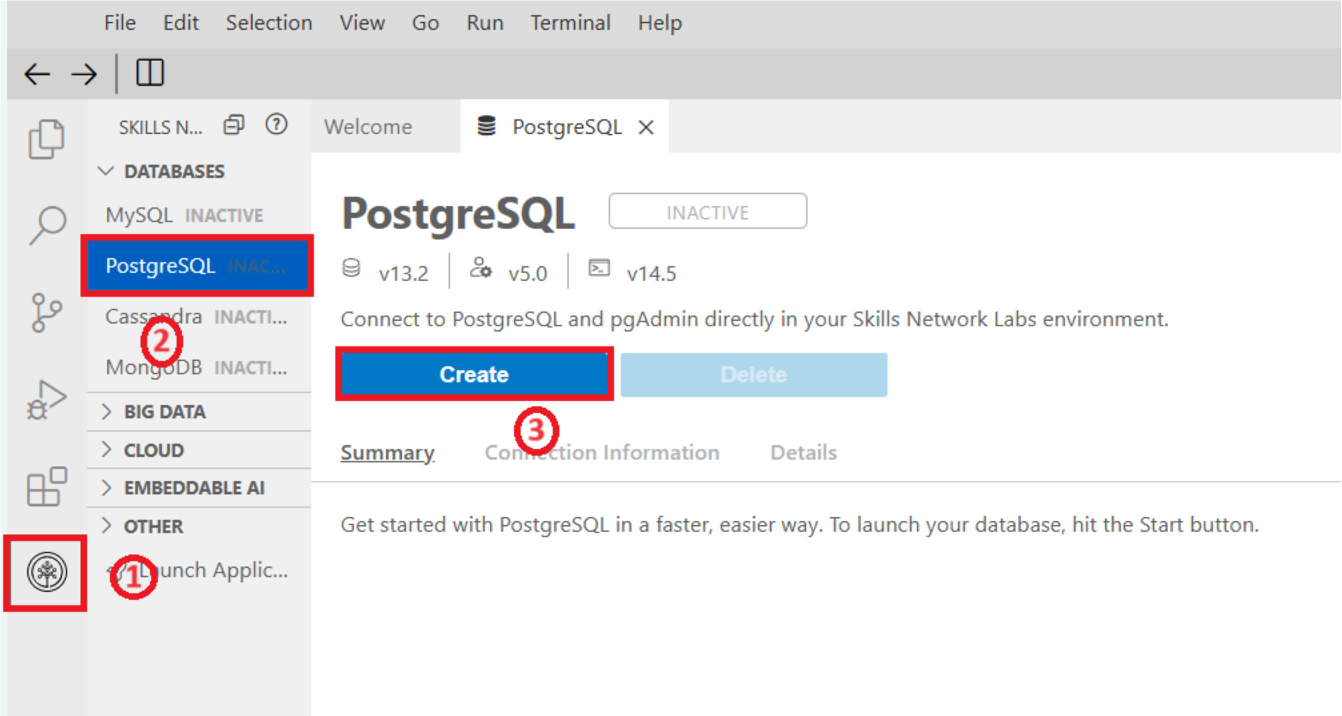
Lab structure

In this exercise, you will go through three tasks to learn how to create and execute views and materialized views in the PostgreSQL database service using the pgAdmin graphical user interface (GUI) tool.

Task A: Restore a database schema and data

To get started with this lab, you will first download the relevant **eBooks** database dump file, then launch PostgreSQL and pgAdmin using the Cloud IDE. You can do this by following these steps:

1. Download the following **eBooks** PostgreSQL dump file (containing the eBooks database schema and data) to your local computer.
   - eBooks_pgsql_dump.tar
2. Click the Skills Network extension button on the left side of the window.
3. Open the **DATABASES** menu and click **PostgreSQL**.
4. Click **Create**. PostgreSQL may take a few moments to start.

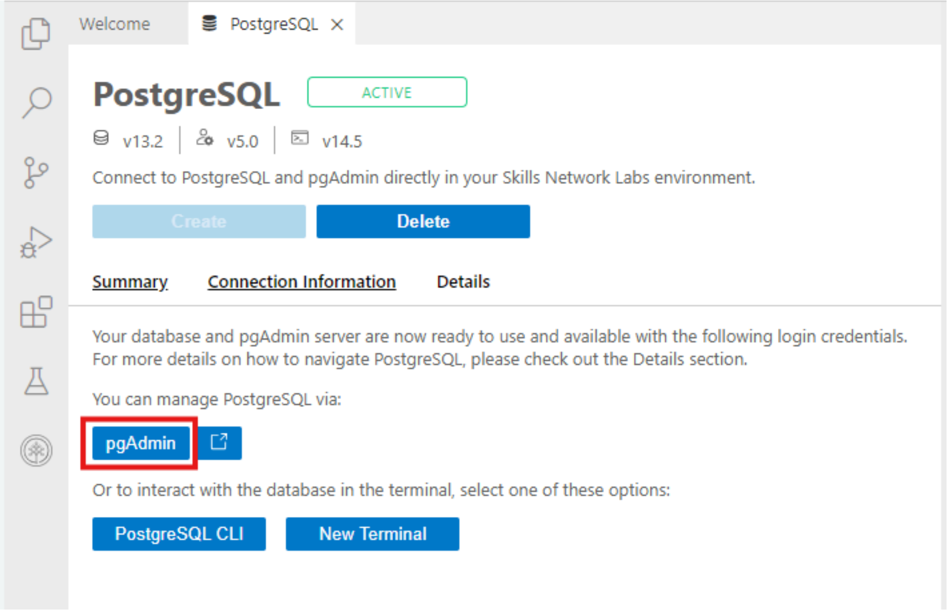


PostgreSQL

Connect to PostgreSQL and pgAdmin directly in your Skills Network Labs environment.

Get started with PostgreSQL in a faster, easier way. To launch your database, hit the Start button.

5. Next, open the pgAdmin Graphical User Interface by clicking **pgAdmin** in the Cloud IDE interface.



PostgreSQL

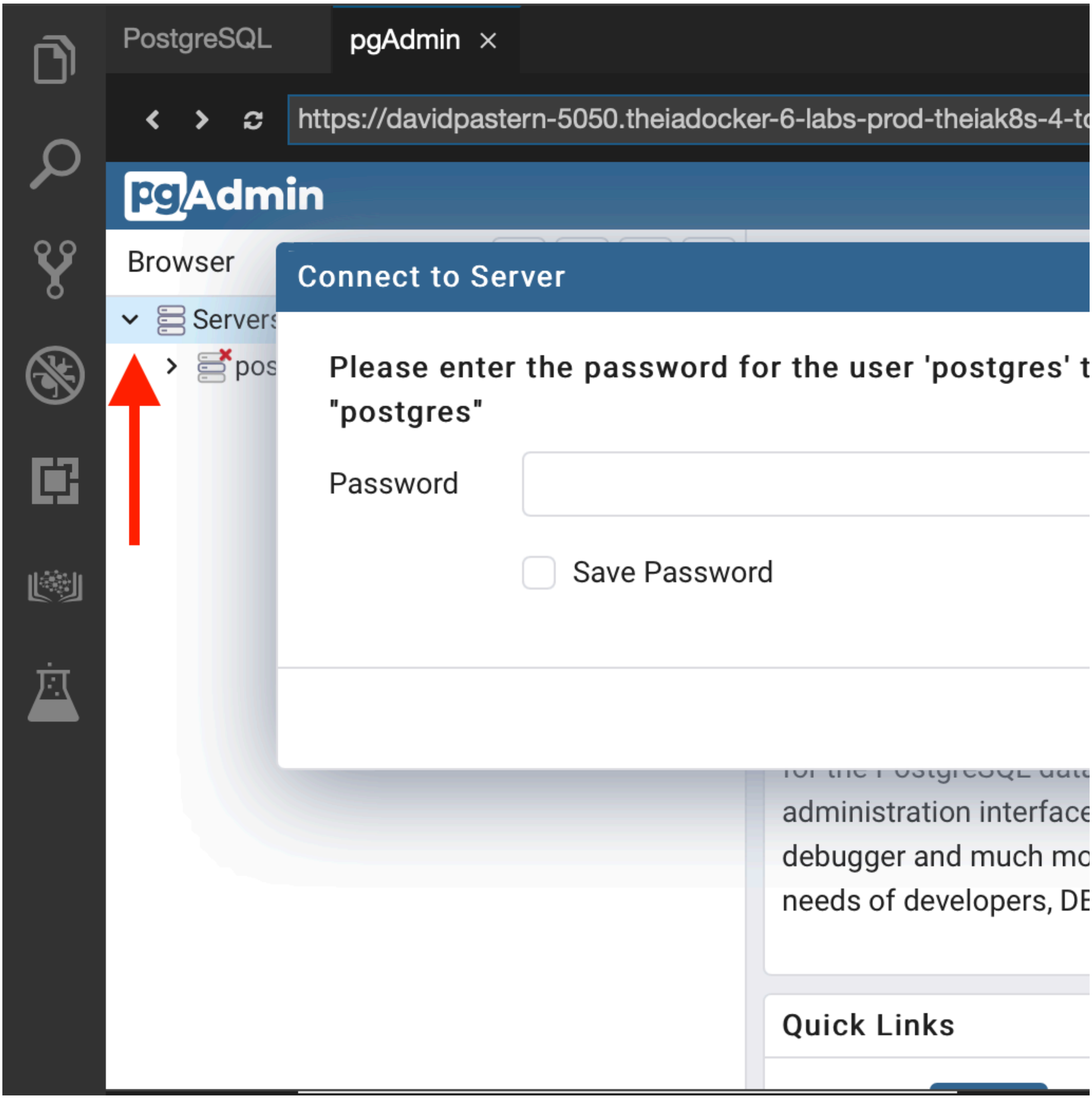
Connect to PostgreSQL and pgAdmin directly in your Skills Network Labs environment.

Your database and pgAdmin server are now ready to use and available with the following login credentials. For more details on how to navigate PostgreSQL, please check out the Details section.

You can manage PostgreSQL via:

Or to interact with the database in the terminal, select one of these options:

6. Once the pgAdmin GUI opens, click **Servers** tab on the left side of the page. You will be prompted to enter a password.



PostgreSQL
pgAdmin ×
https://davidpastern-5050.theiadocker-6-labs-prod-theiak8s-4-t
pgAdmin
Browser
Servers
pos

Connect to Server

Please enter the password for the user 'postgres' t
"postgres"

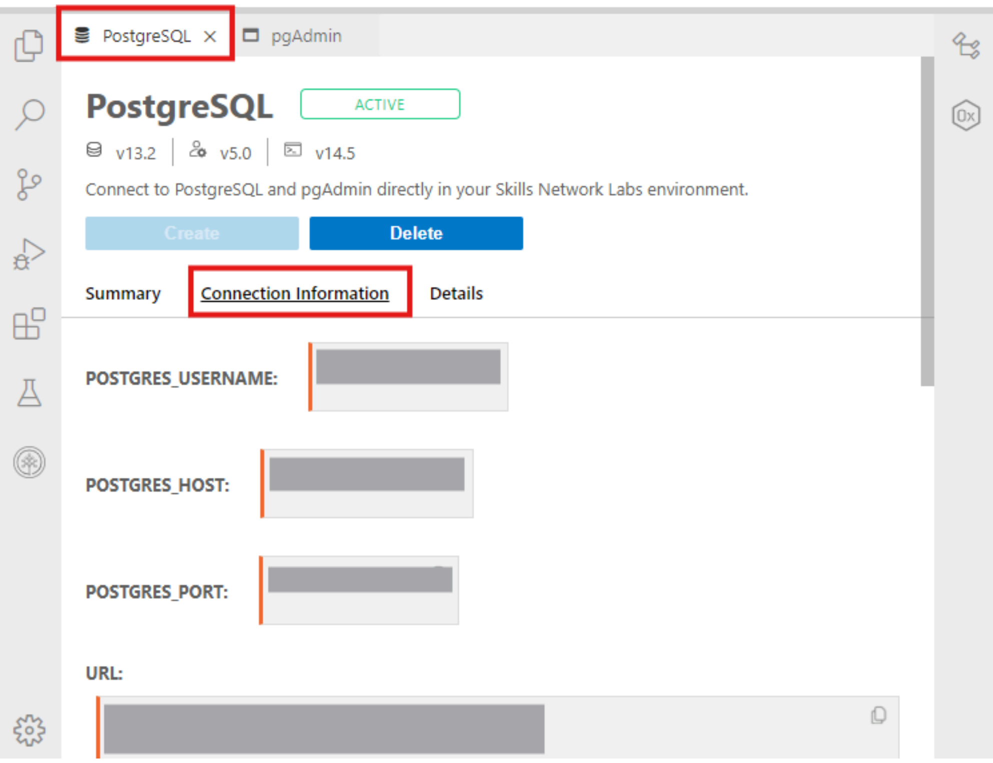
Password

□ Save Password

administration interface
debugger and much mo
needs of developers, DE

Quick Links

7. To retrieve your password, click **PostgreSQL** tab near the top of the interface and select **Connection Information** tab.



PostgreSQL

Connect to PostgreSQL and pgAdmin directly in your Skills Network Labs environment.

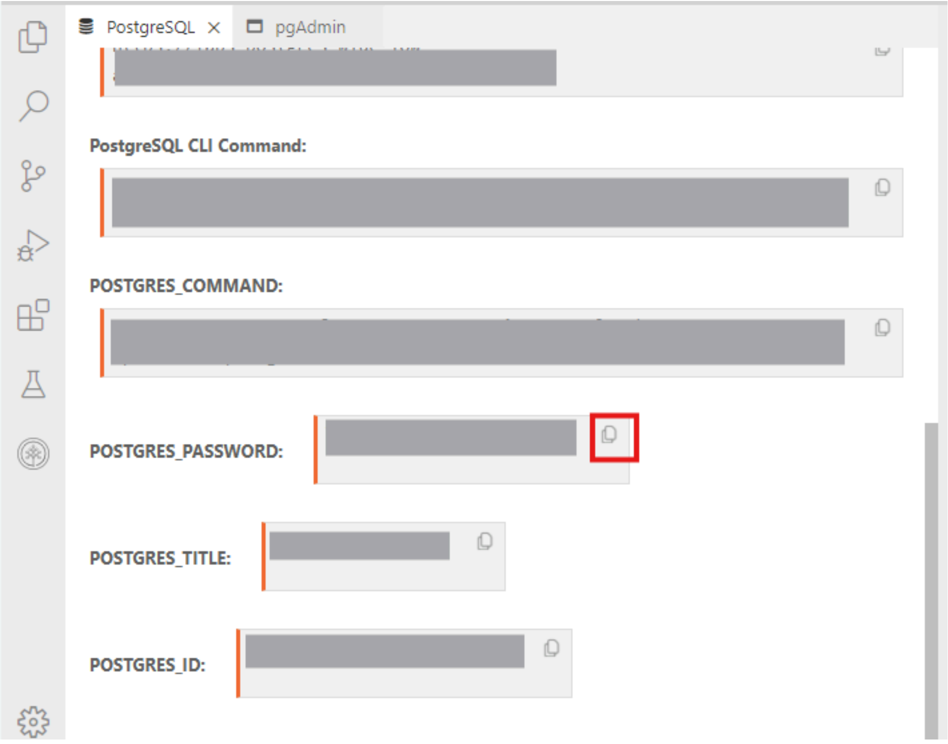
POSTGRES_USERNAME: [redacted]

POSTGRES_HOST: [redacted]

POSTGRES_PORT: [redacted]

URL: [redacted]

8. Scroll down and click the Copy icon on the left of your password to copy the session password onto your clipboard.



[redacted]

PostgreSQL CLI Command: [redacted]

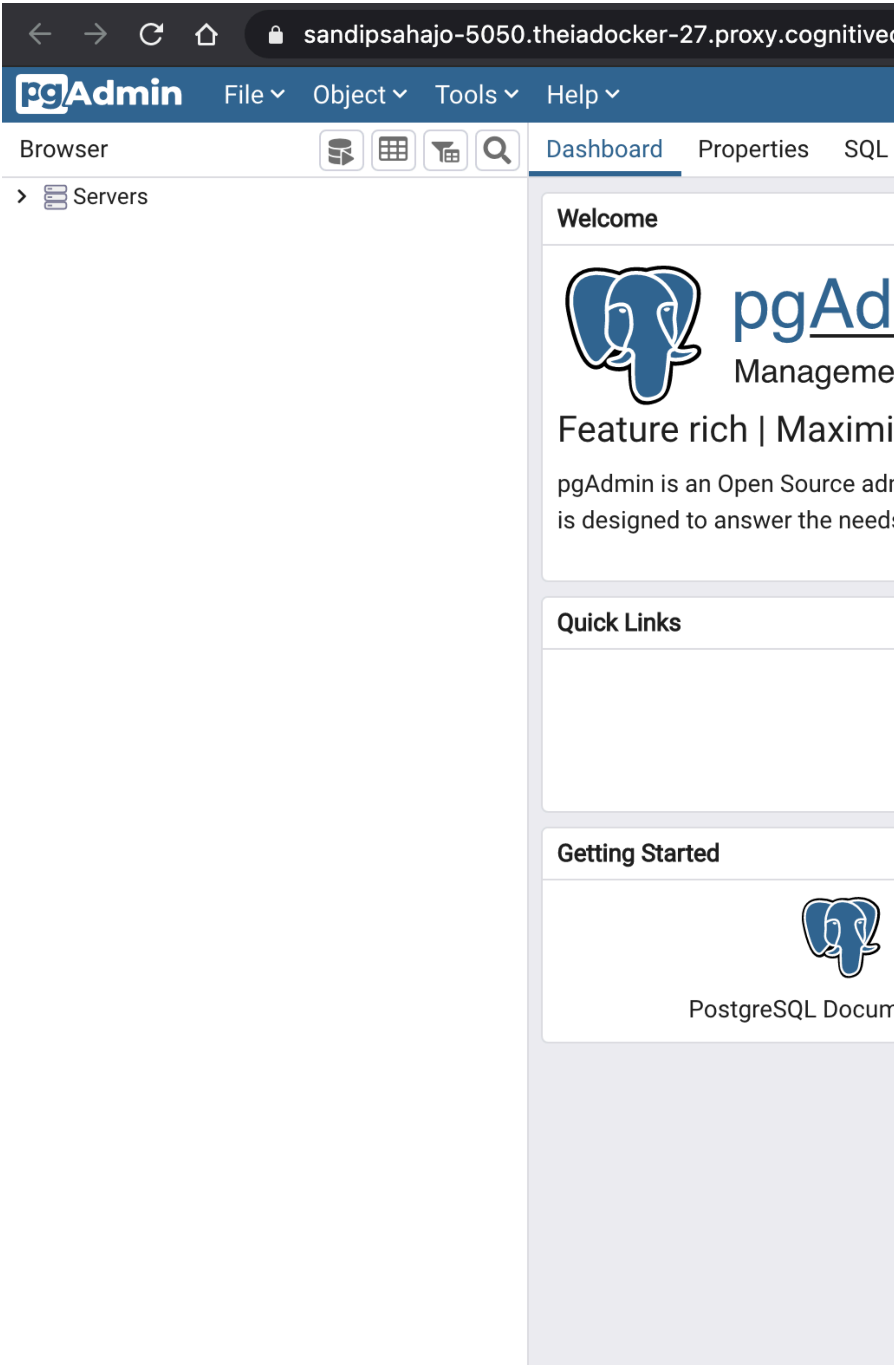
POSTGRES_COMMAND: [redacted]

POSTGRES_PASSWORD: [redacted]

POSTGRES_TITLE: [redacted]

POSTGRES_ID: [redacted]

9. Navigate back to the **pgAdmin** tab and paste your password, then click **OK**.
10. You will then be able to access the pgAdmin GUI tool.



Welcome

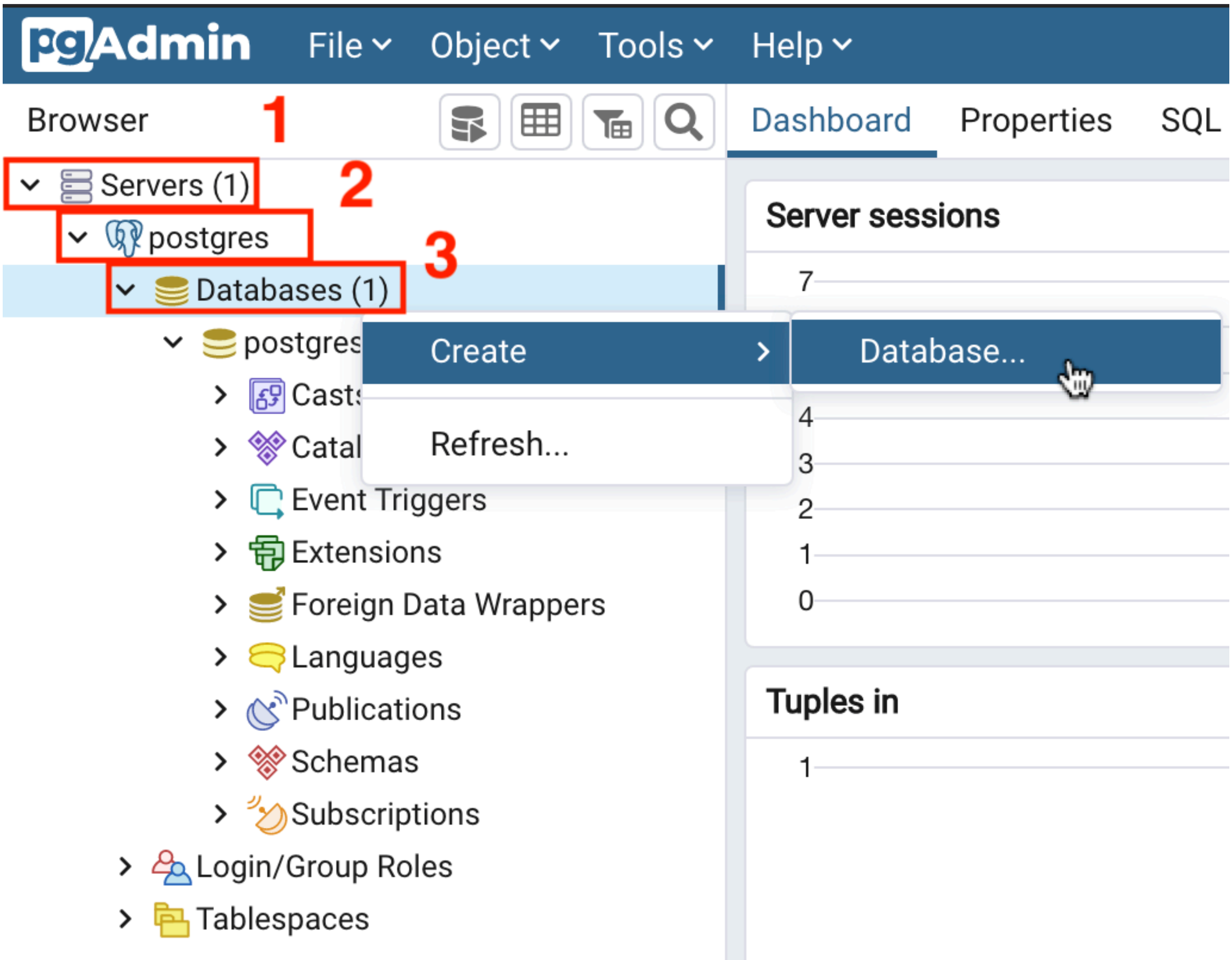
Feature rich | Maximi

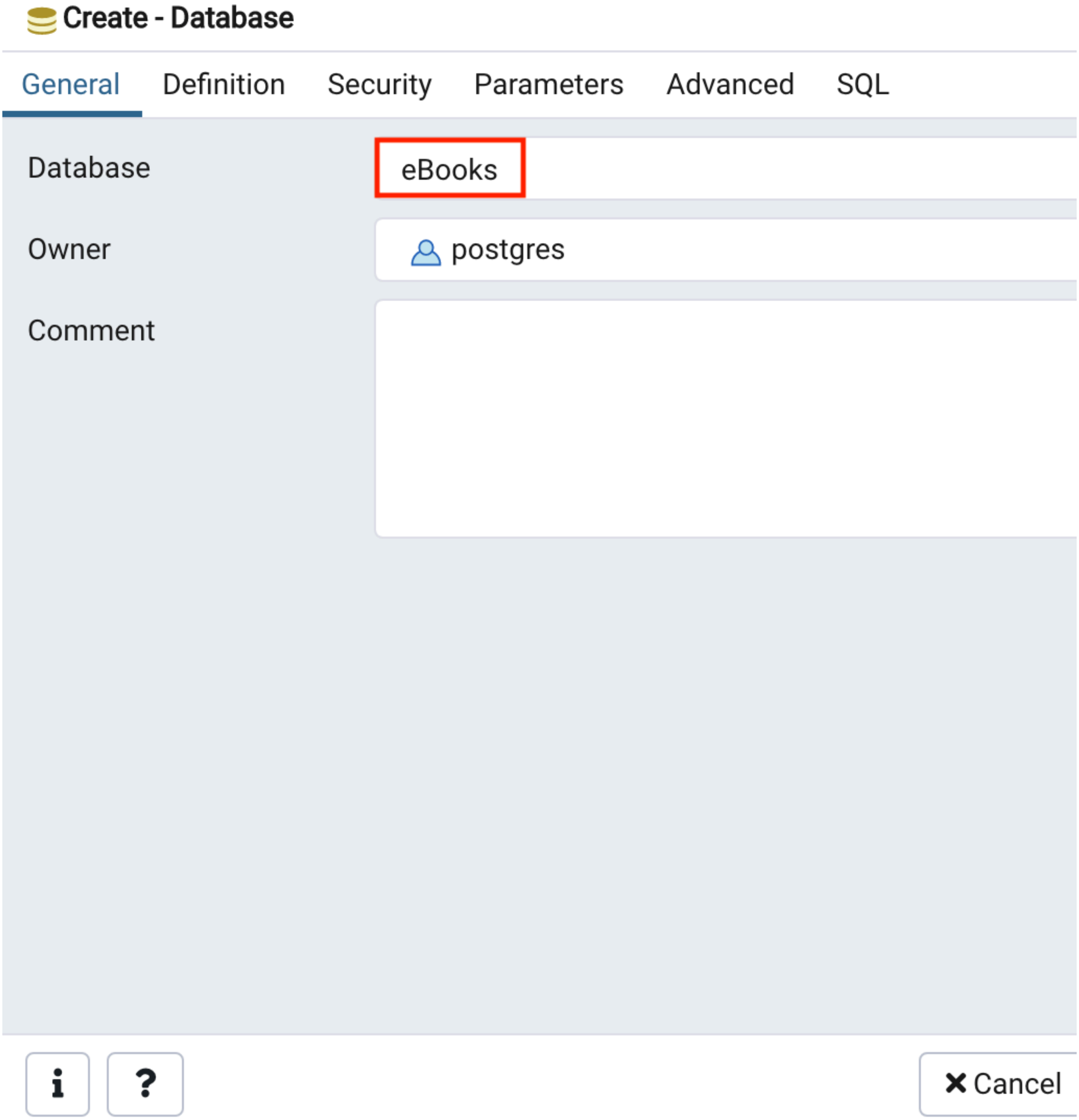
pgAdmin is an Open Source adr is designed to answer the need:

Quick Links

Getting Started

11. In the tree view, expand **Servers** > **postgres** > **Databases**. Enter your PostgreSQL service session password if prompted during the process. Right-click on **Databases** and go to **Create > Database**. Type **eBooks** as the database name and click **Save**.





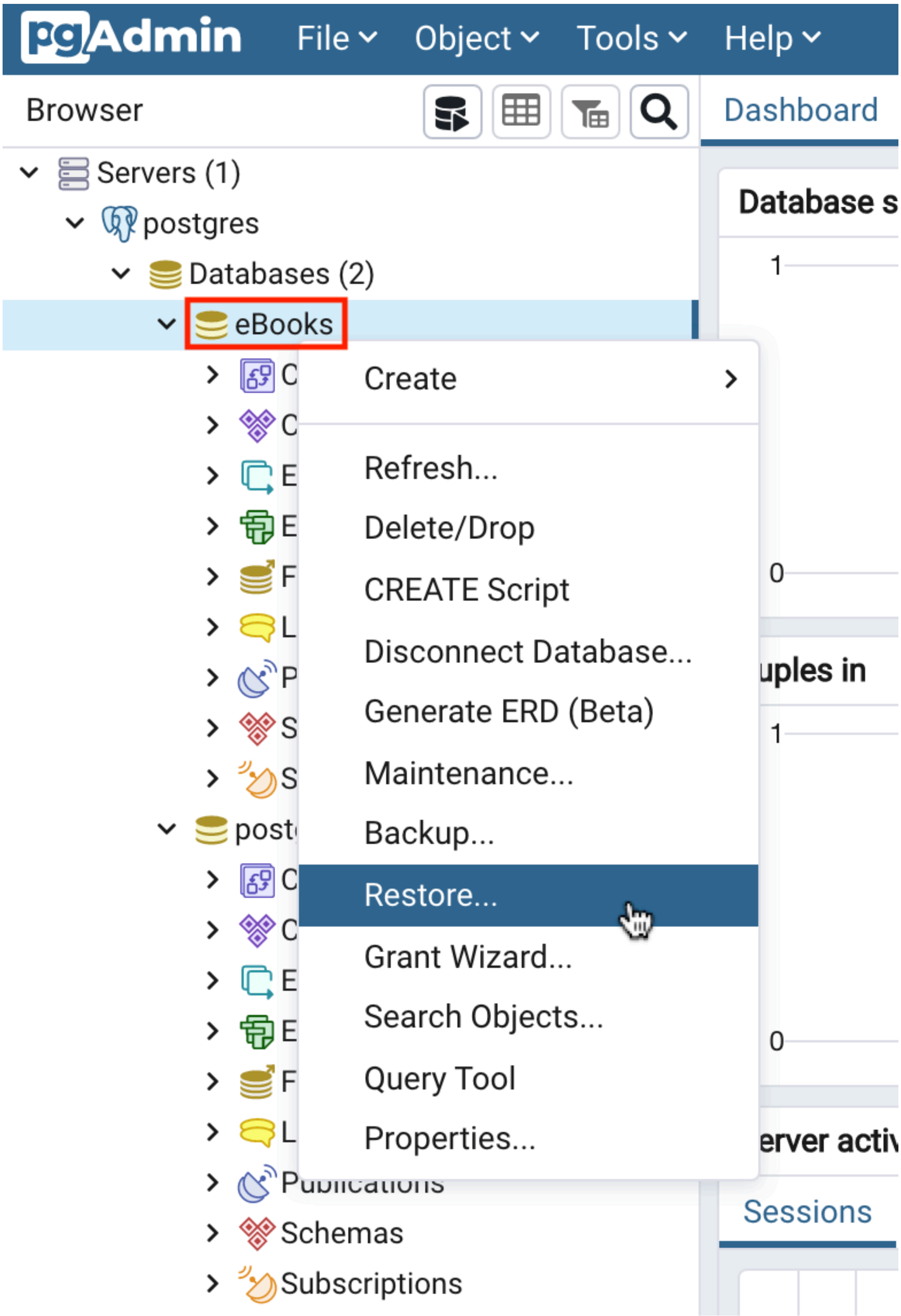
Create - Database

Database: eBooks

Owner: postgres

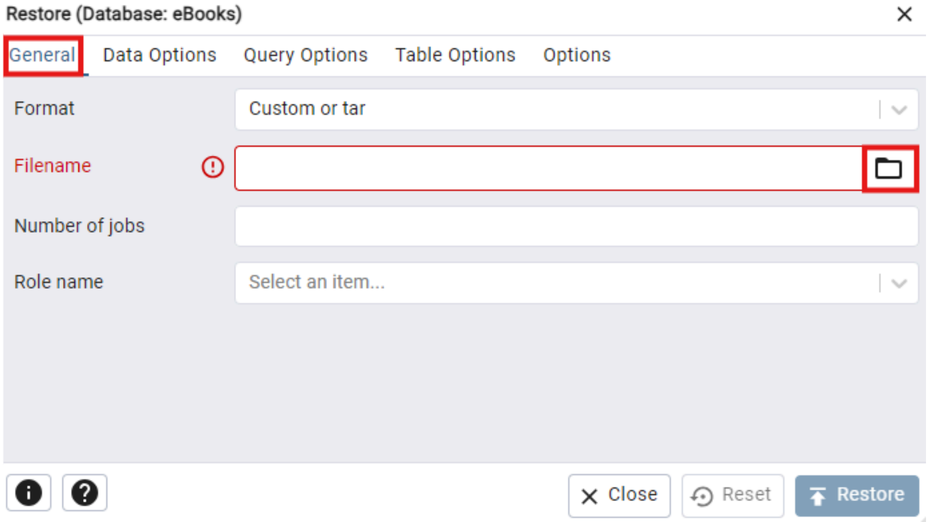
Comment

12. In the tree-view, expand **eBooks**. Right-click **eBooks** and select **Restore**.



13. Follow the instructions below to restore and proceed to Task B:

   - On the **General** tab, click **Select file** by the **Filename** box.


Restore (Database: eBooks)

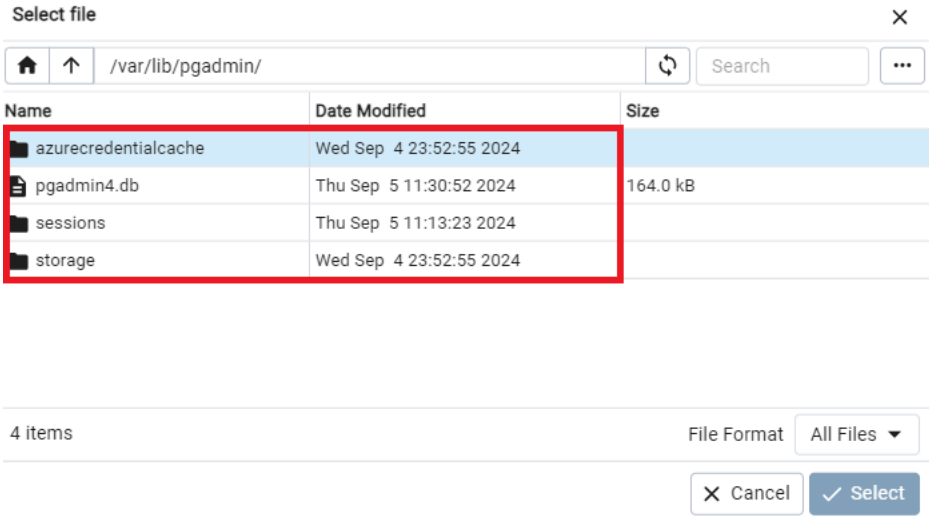
Format: Custom or tar

Filename:

Number of jobs:

Role name: Select an item...

- Ensure that you upload the files to this path: /var/lib/pgadmin/. To do this, you can either manually navigate to the path (or) copy /var/lib/pgadmin/, replace /home/ with it, and press Enter. You should then see some default files in that path, as shown below.


Select file

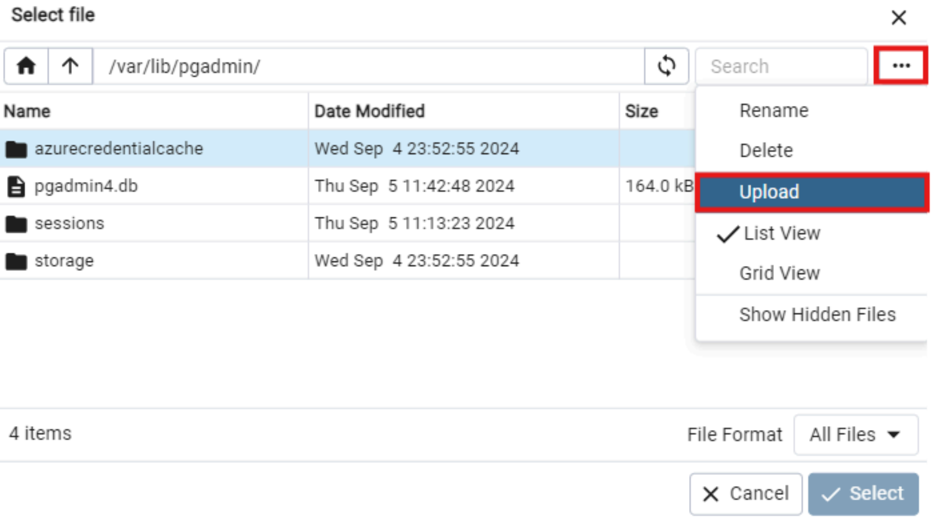
/var/lib/pgadmin/

Name	Date Modified	Size
azurecredentialcache	Wed Sep 4 23:52:55 2024	
pgadmin4.db	Thu Sep 5 11:30:52 2024	164.0 kB
sessions	Thu Sep 5 11:13:23 2024	
storage	Wed Sep 4 23:52:55 2024	

4 items

File Format: All Files

- Click on the three dots, then select **Upload**.


Select file

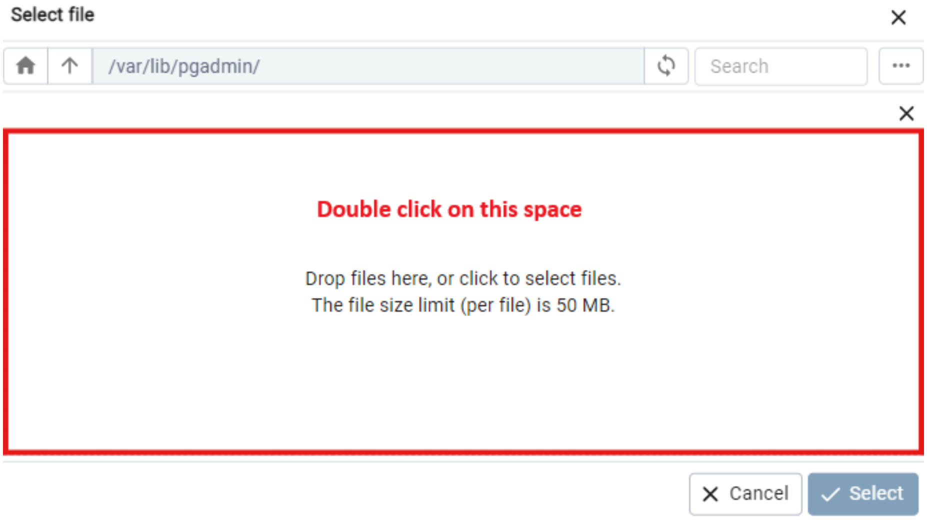
/var/lib/pgadmin/

Name	Date Modified	Size
azurecredentialcache	Wed Sep 4 23:52:55 2024	
pgadmin4.db	Thu Sep 5 11:42:48 2024	164.0 kB
sessions	Thu Sep 5 11:13:23 2024	
storage	Wed Sep 4 23:52:55 2024	

4 items

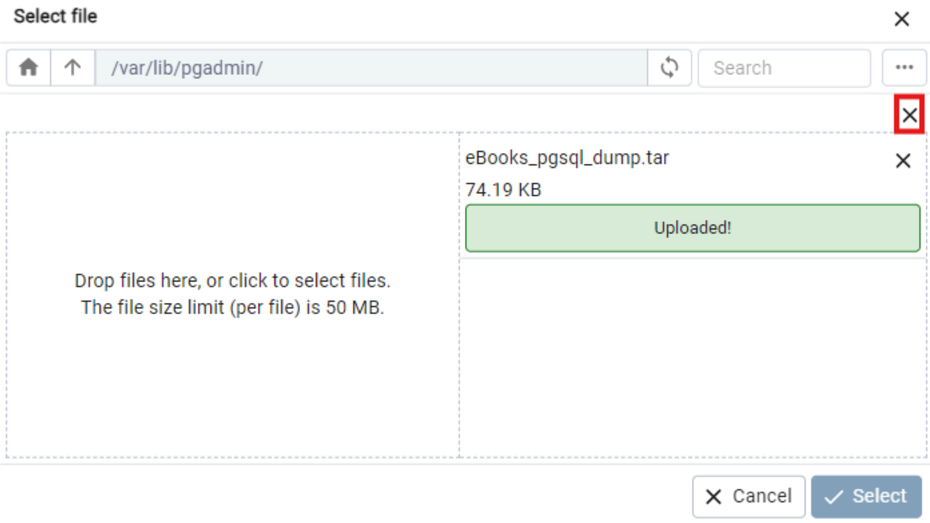
File Format: All Files

- Double-click on the drop files area and load the **eBooks_pgsql_dump.tar** you downloaded earlier on your local computer.



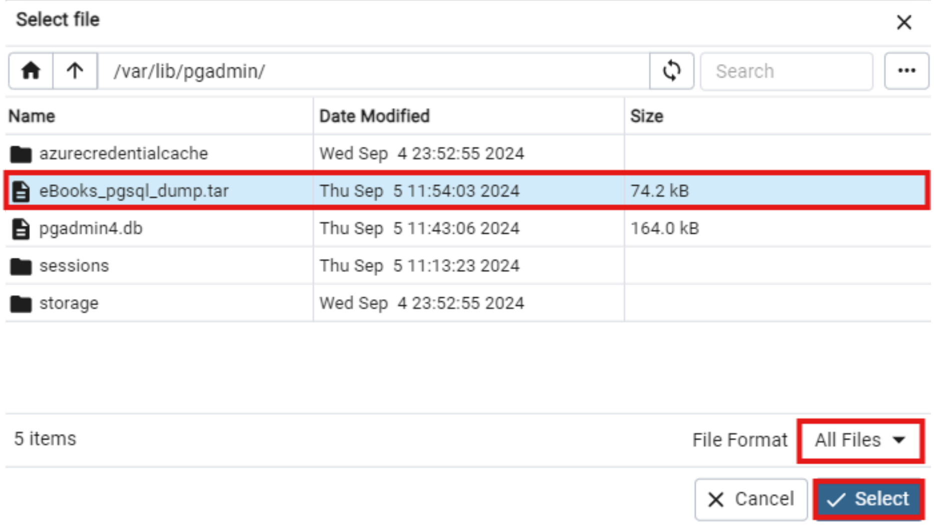
Drop files here, or click to select files.
The file size limit (per file) is 50 MB.

- When the upload is complete, close the drop files area by clicking **X**.



Drop files here, or click to select files.
The file size limit (per file) is 50 MB.

- Ensure **Format** is set to **All Files**, select the uploaded **eBooks_pgsql_dump.tar** file from the list, and then click **Select**.

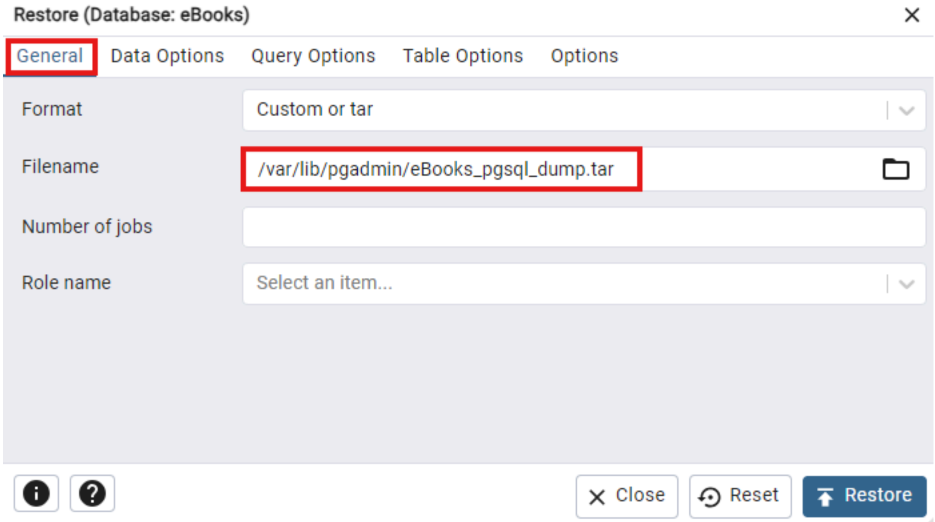


Name	Date Modified	Size
azurecredentialcache	Wed Sep 4 23:52:55 2024	
eBooks_pgsql_dump.tar	Thu Sep 5 11:54:03 2024	74.2 kB
pgadmin4.db	Thu Sep 5 11:43:06 2024	164.0 kB
sessions	Thu Sep 5 11:13:23 2024	
storage	Wed Sep 4 23:52:55 2024	

5 items

File Format: All Files

- In the General tab, ensure the filename path matches the one shown below. If you see a different path that includes “None,” modify it accordingly.



Restore (Database: eBooks)

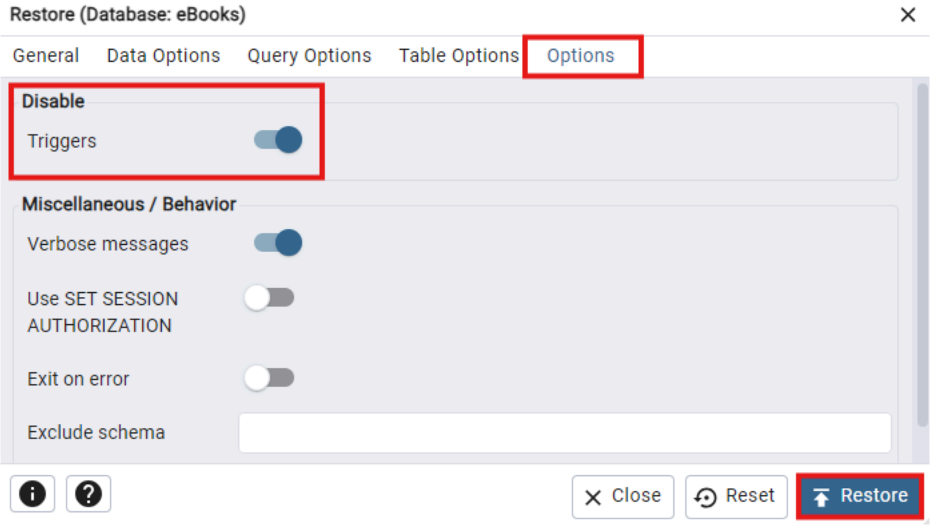
Format: Custom or tar

Filename: /var/lib/pgadmin/eBooks_pgsql_dump.tar

Number of jobs:

Role name: Select an item...

- Now switch to the **Options** tab. Under **Disable**, toggle on the **Triggers** option, and then click Restore.



Restore (Database: eBooks)

Disable

Triggers: on

Miscellaneous / Behavior

Verbose messages: on

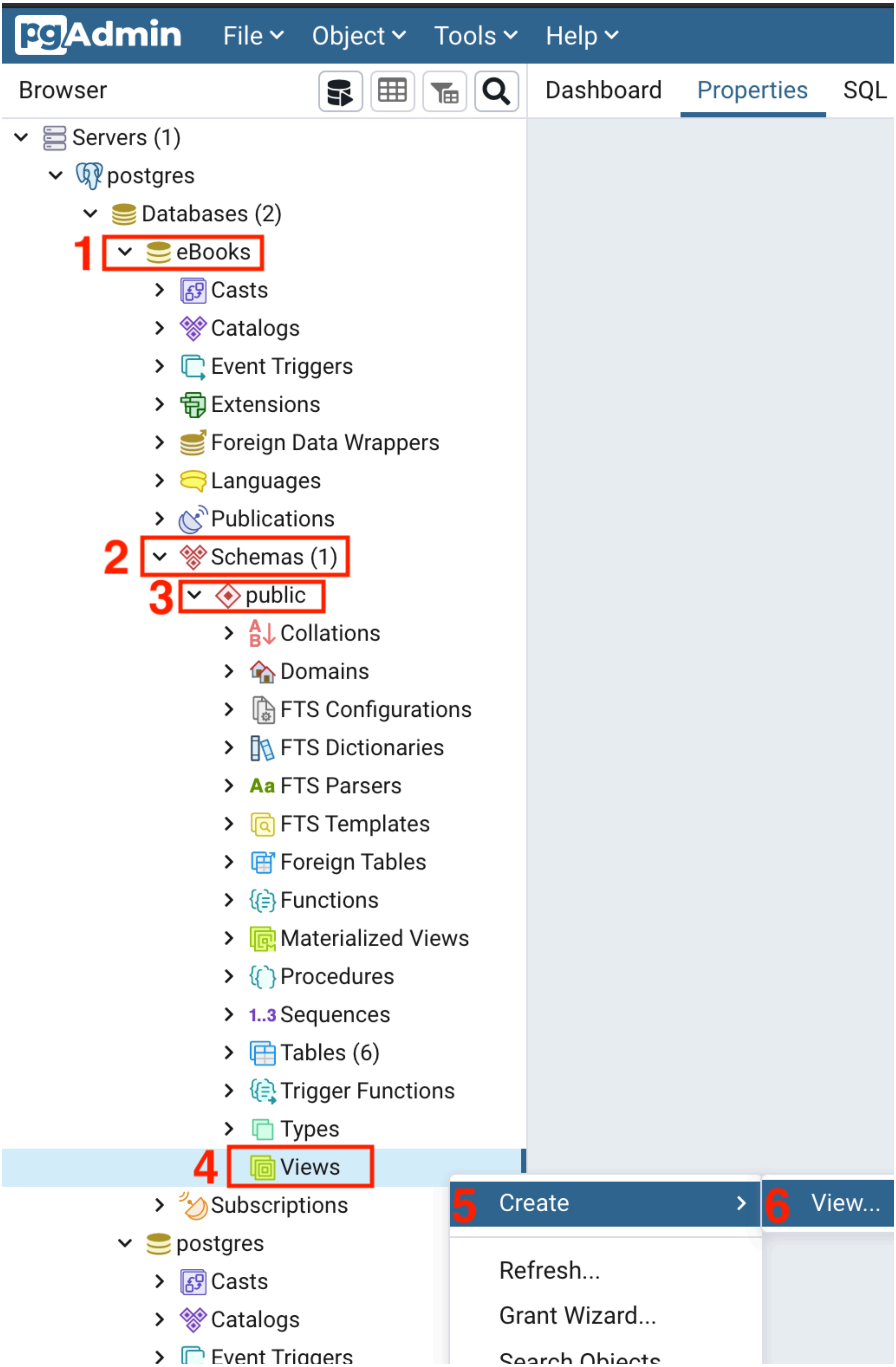
Use SET SESSION AUTHORIZATION: off

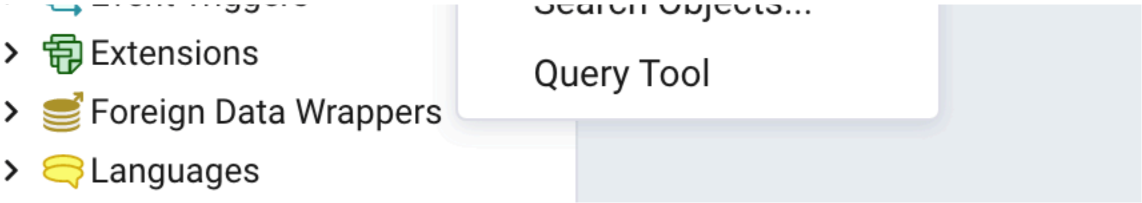
Exit on error: off

Exclude schema:

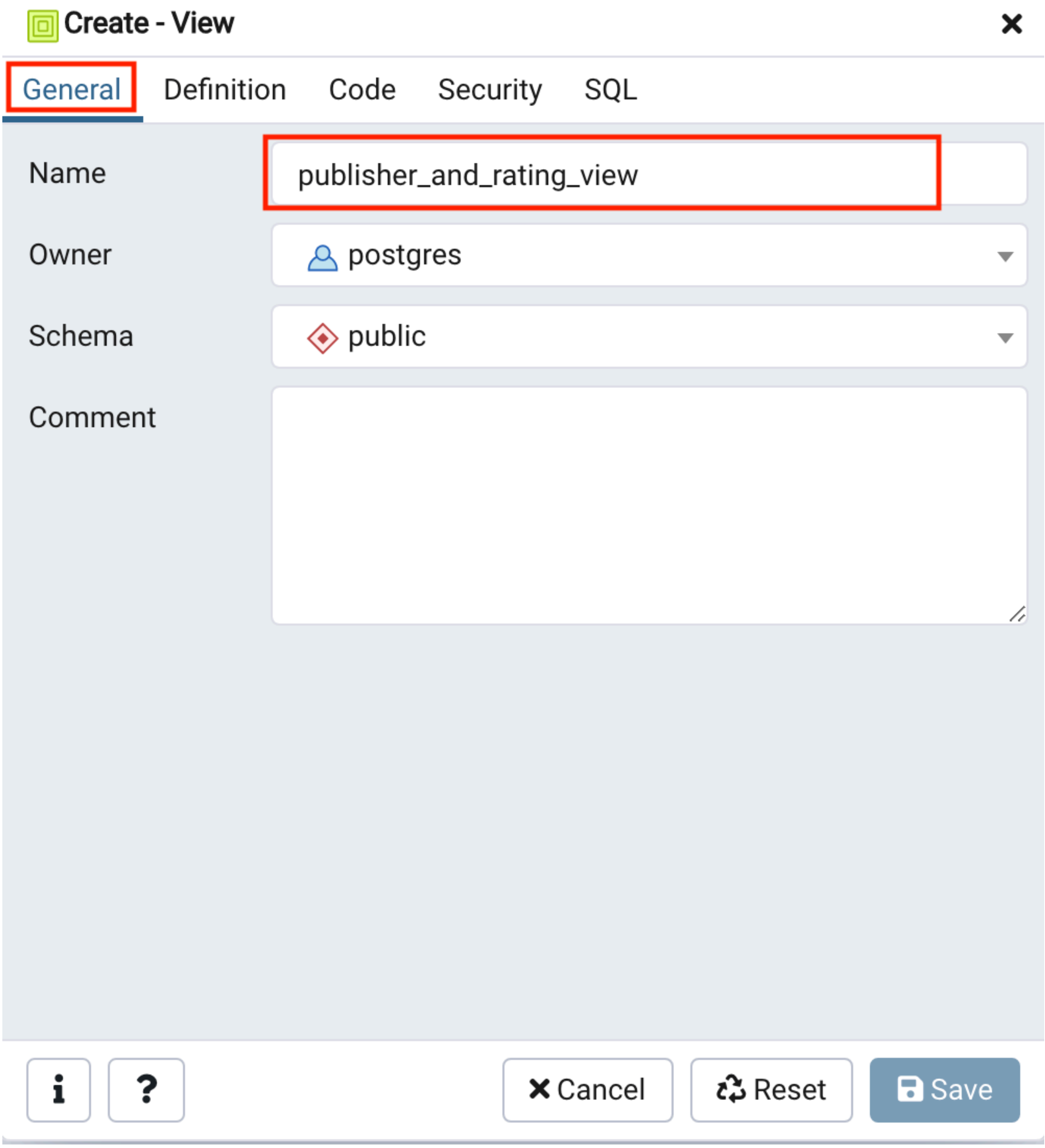
Task B: Create and execute a view

1. In the tree-view, expand **eBooks** > **Schemas** > **public**. Right-click **Views** and go to **Create** > **View**.


pgAdmin
File
Object
Tools
Help
Browser
Dashboard
Properties
SQL
Servers (1)
postgres
Databases (2)
1
eBooks
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
2
Schemas (1)
3
public
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Procedures
Sequences
Tables (6)
Trigger Functions
Types
4
Views
Subscriptions
postgres
Casts
Catalogs
Event Triggers
5
Create
6
View...
Refresh...
Grant Wizard...
Search Objects



2. On the **General** tab, type **publisher_and_rating_view** as the name of the view. Then, switch to the **Code** tab.



Name: publisher_and_rating_view

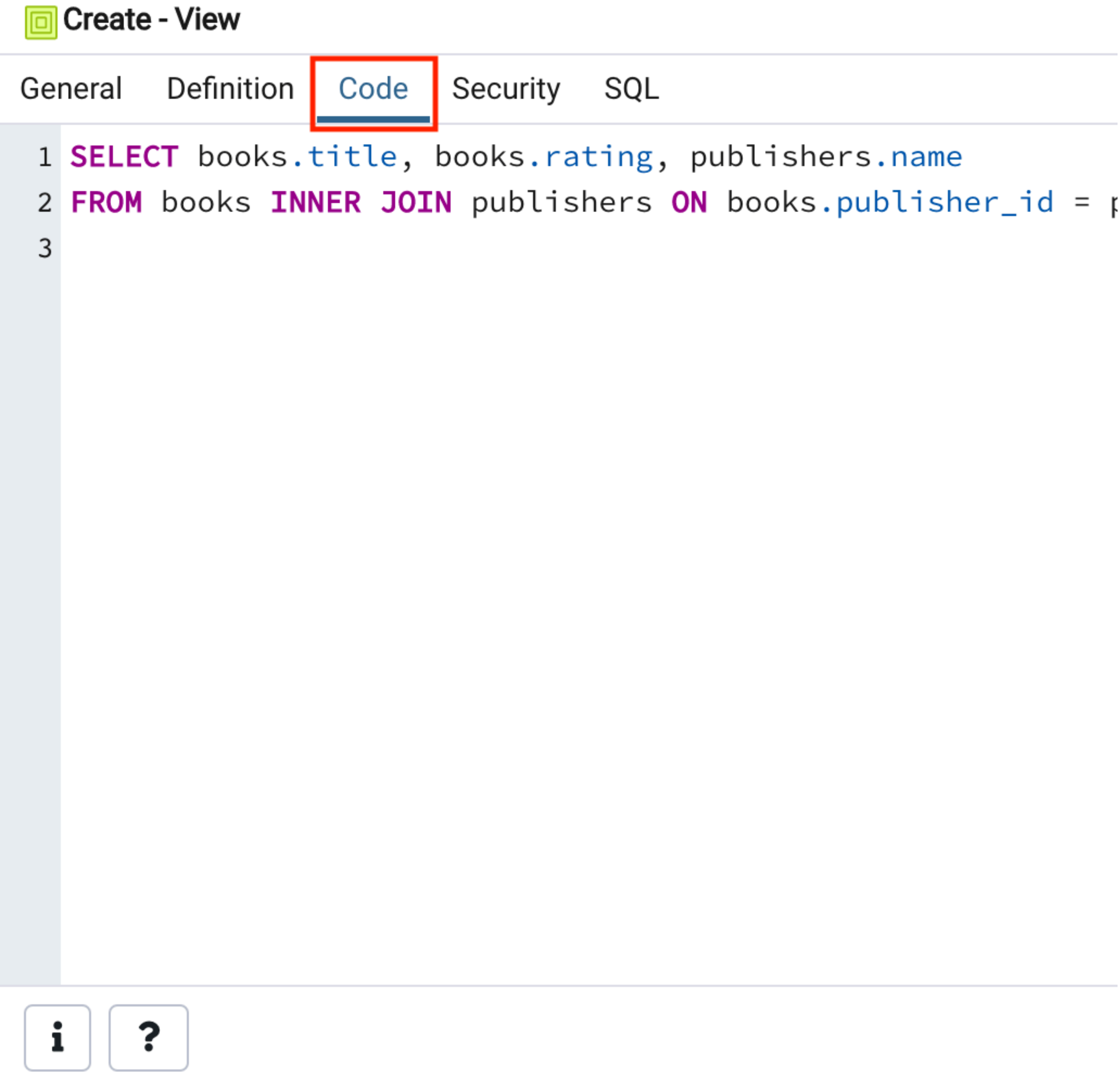
Owner: postgres

Schema: public

Comment:

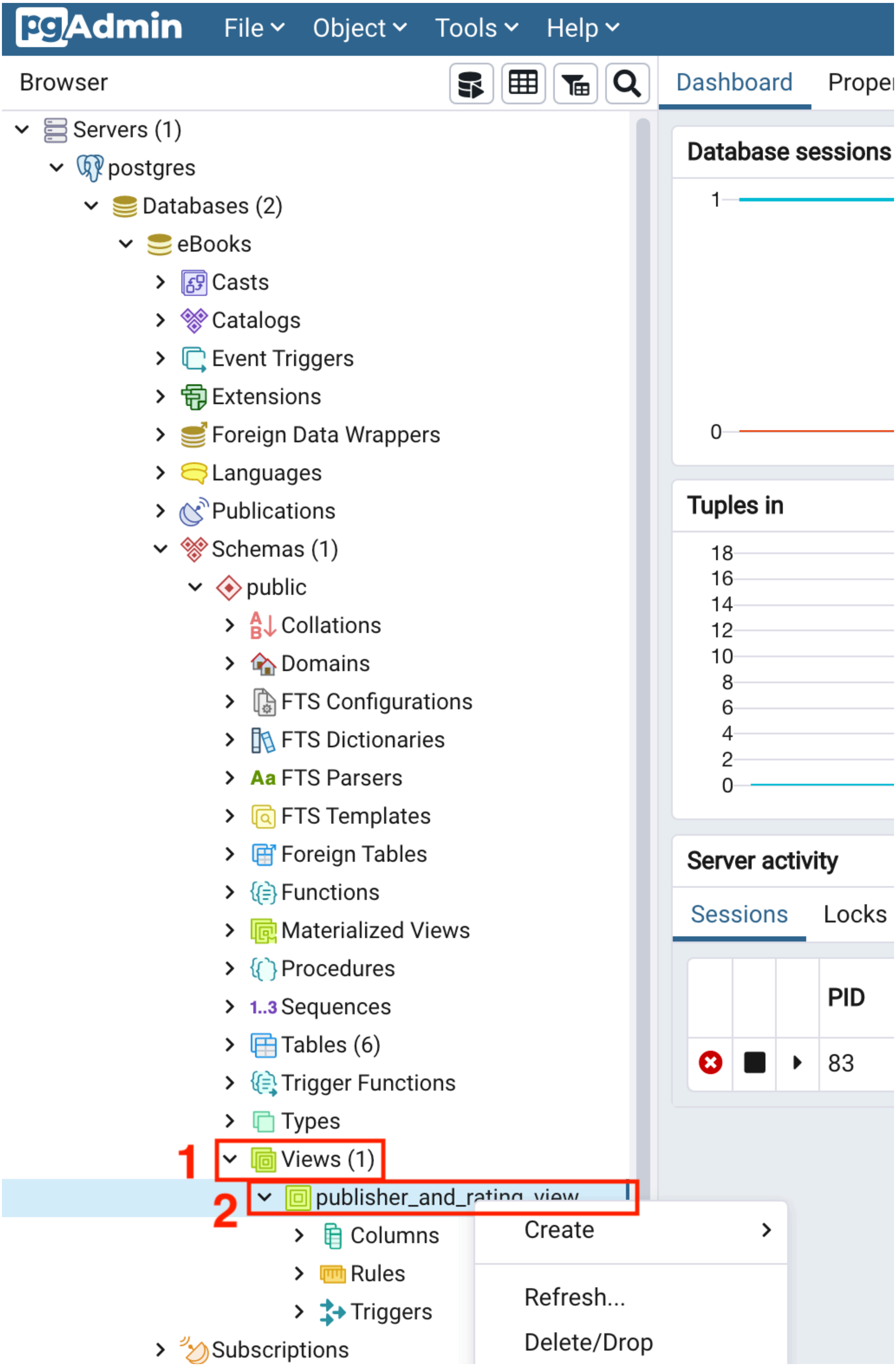
3. On the **Code** tab, copy and paste the following code. Then click **Save**.

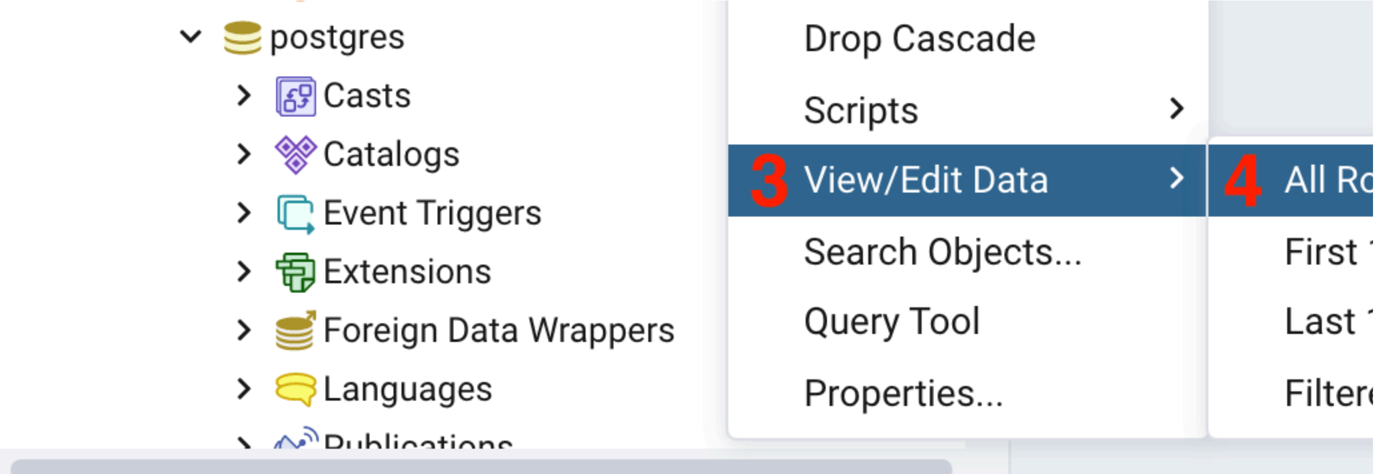
```
SELECT books.title, books.rating, publishers.name
FROM books INNER JOIN publishers ON books.publisher_id = publishers.publisher_id
```



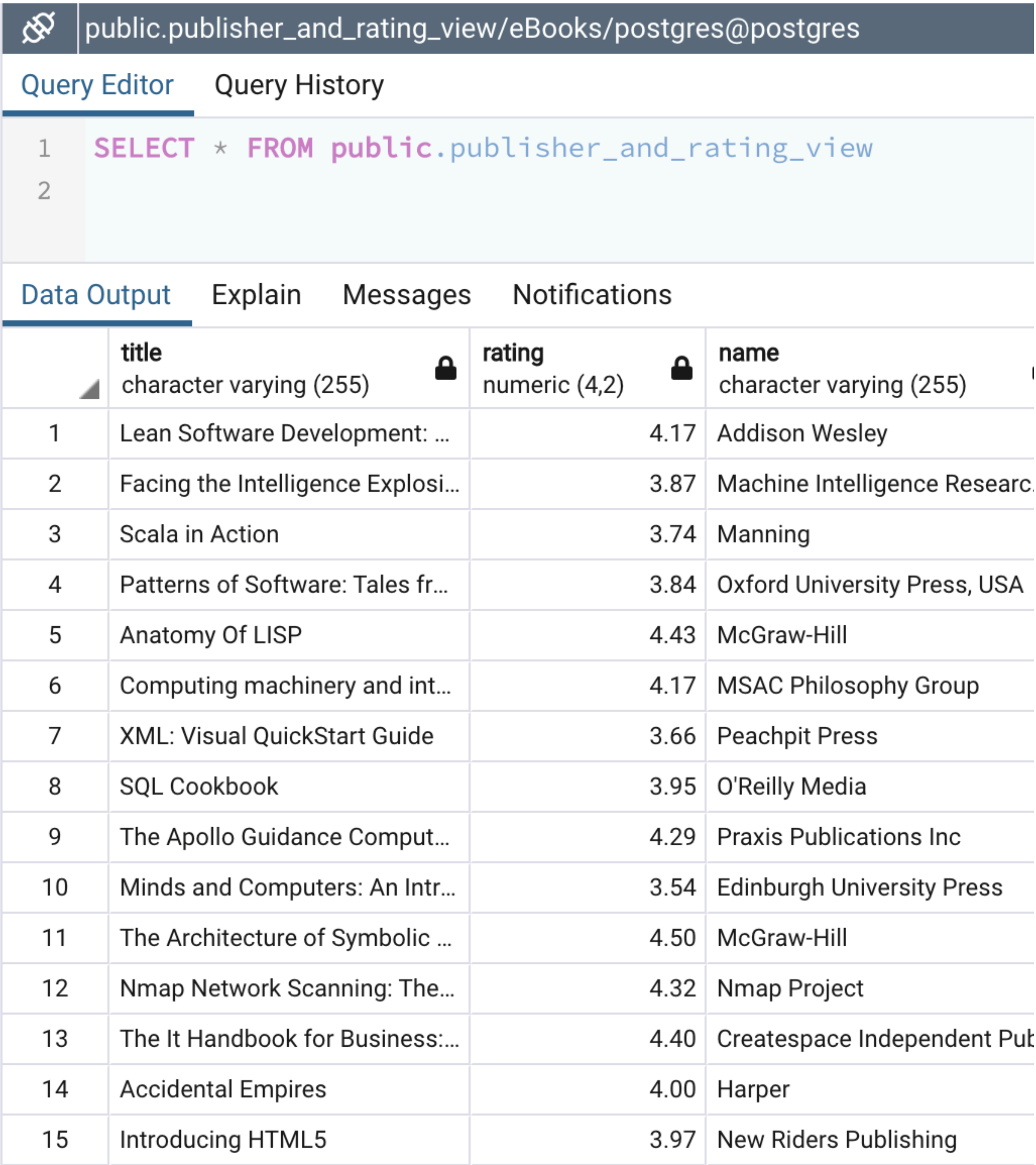
```
SELECT books.title, books.rating, publishers.name
FROM books INNER JOIN publishers ON books.publisher_id = p
```

4. In the tree view, expand **Views**. Right-click **publisher_and_rating_view** and go to **View/Edit Data > All Rows**.


pgAdmin
File
Object
Tools
Help
Browser
Dashboard
Prope
Servers (1)
postgres
Databases (2)
eBooks
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas (1)
public
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Procedures
Sequences
Tables (6)
Trigger Functions
Types
1
Views (1)
2
publisher_and_rating_view
Columns
Rules
Triggers
Subscriptions
Database sessions
1
0
Tuples in
18
16
14
12
10
8
6
4
2
0
Server activity
Sessions
Locks
PID
83
Create
Refresh...
Delete/Drop



5. You will access the view you created. This action allows you to access and view the tables in your database.

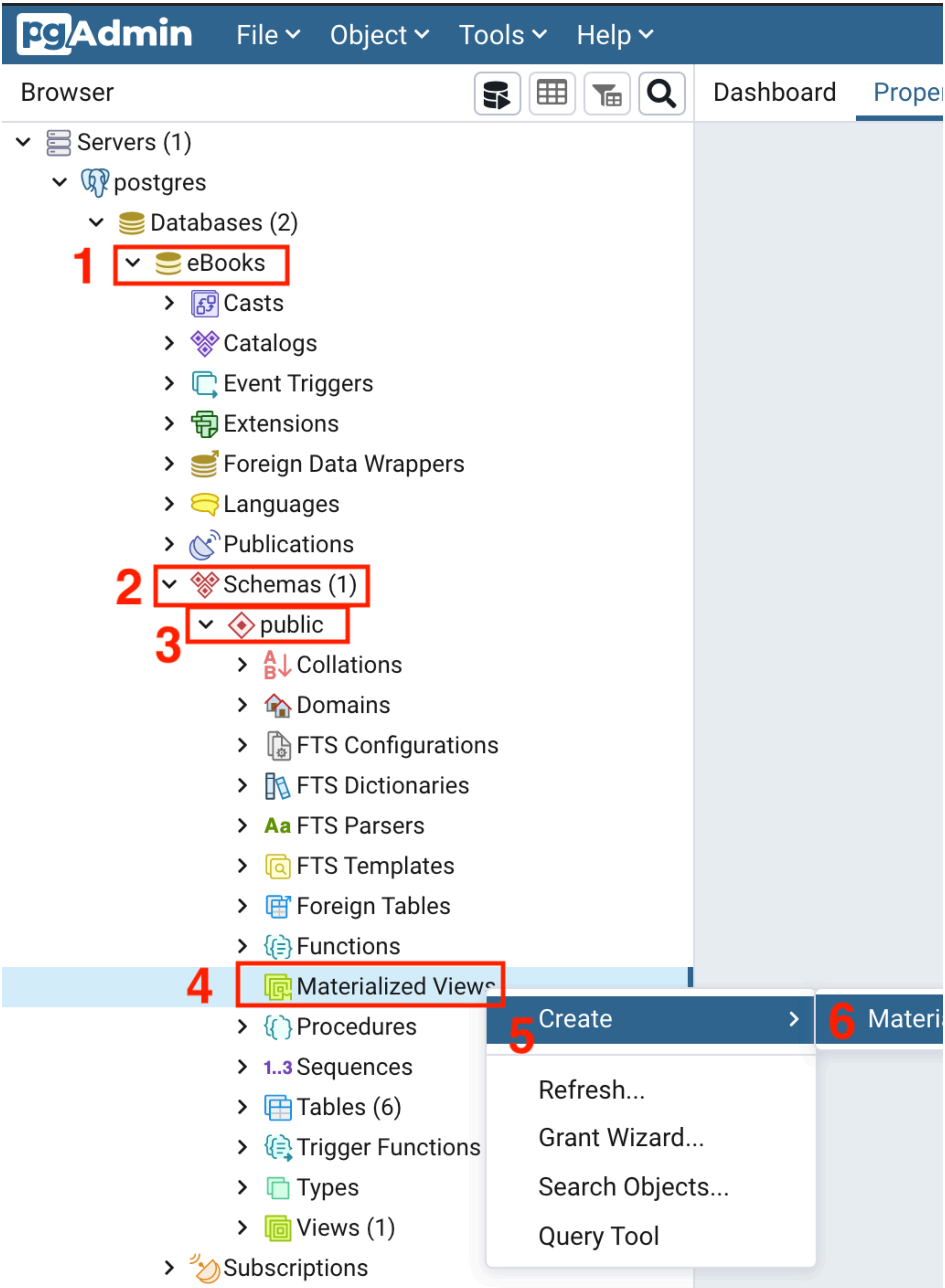


```
SELECT * FROM public.publisher_and_rating_view
```

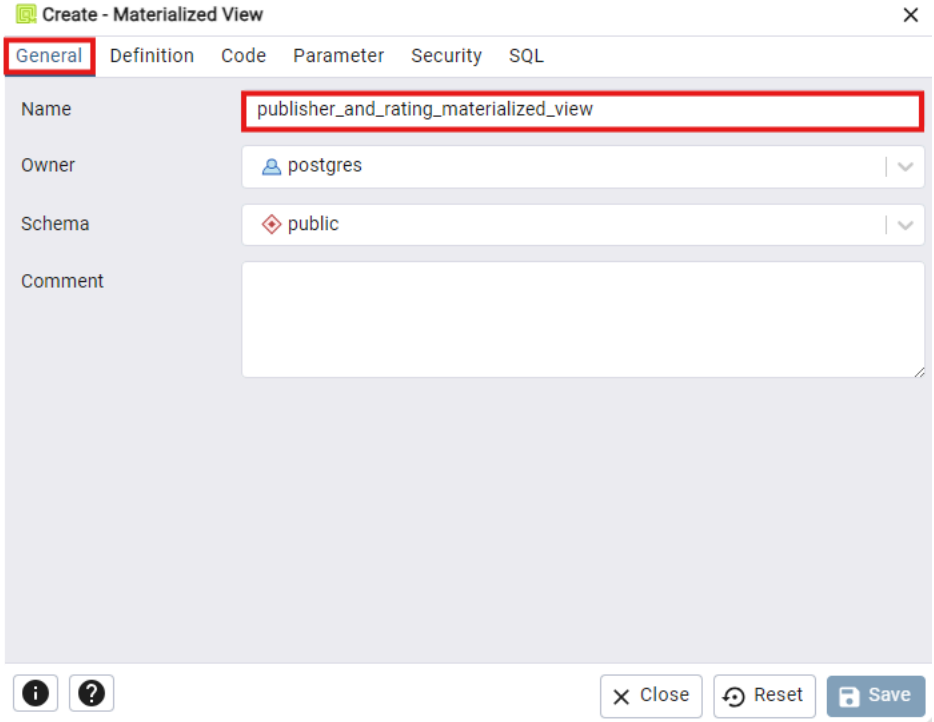
	title character varying (255)	rating numeric (4,2)	name character varying (255)
1	Lean Software Development: ...	4.17	Addison Wesley
2	Facing the Intelligence Explosi...	3.87	Machine Intelligence Researc
3	Scala in Action	3.74	Manning
4	Patterns of Software: Tales fr...	3.84	Oxford University Press, USA
5	Anatomy Of LISP	4.43	McGraw-Hill
6	Computing machinery and int...	4.17	MSAC Philosophy Group
7	XML: Visual QuickStart Guide	3.66	Peachpit Press
8	SQL Cookbook	3.95	O'Reilly Media
9	The Apollo Guidance Comput...	4.29	Praxis Publications Inc
10	Minds and Computers: An Intr...	3.54	Edinburgh University Press
11	The Architecture of Symbolic ...	4.50	McGraw-Hill
12	Nmap Network Scanning: The...	4.32	Nmap Project
13	The It Handbook for Business:...	4.40	Createspace Independent Pub
14	Accidental Empires	4.00	Harper
15	Introducing HTML5	3.97	New Riders Publishing

Task C: Create and execute a materialized view

1. In the tree view, expand **eBooks > Schemas > public**. Right-click **Materialized Views** and go to **Create > Materialized View**.



2. On the **General** tab, type **publisher_and_rating_materialized_view** as name of the view. Then switch to the **Code** tab.



Name: publisher_and_rating_materialized_view

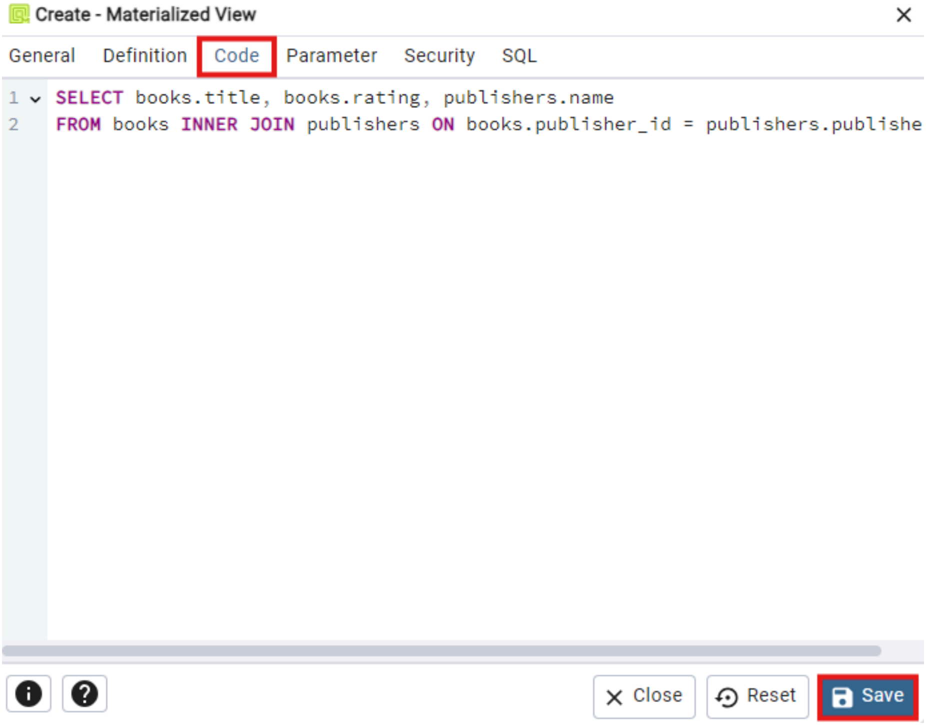
Owner: postgres

Schema: public

Comment:

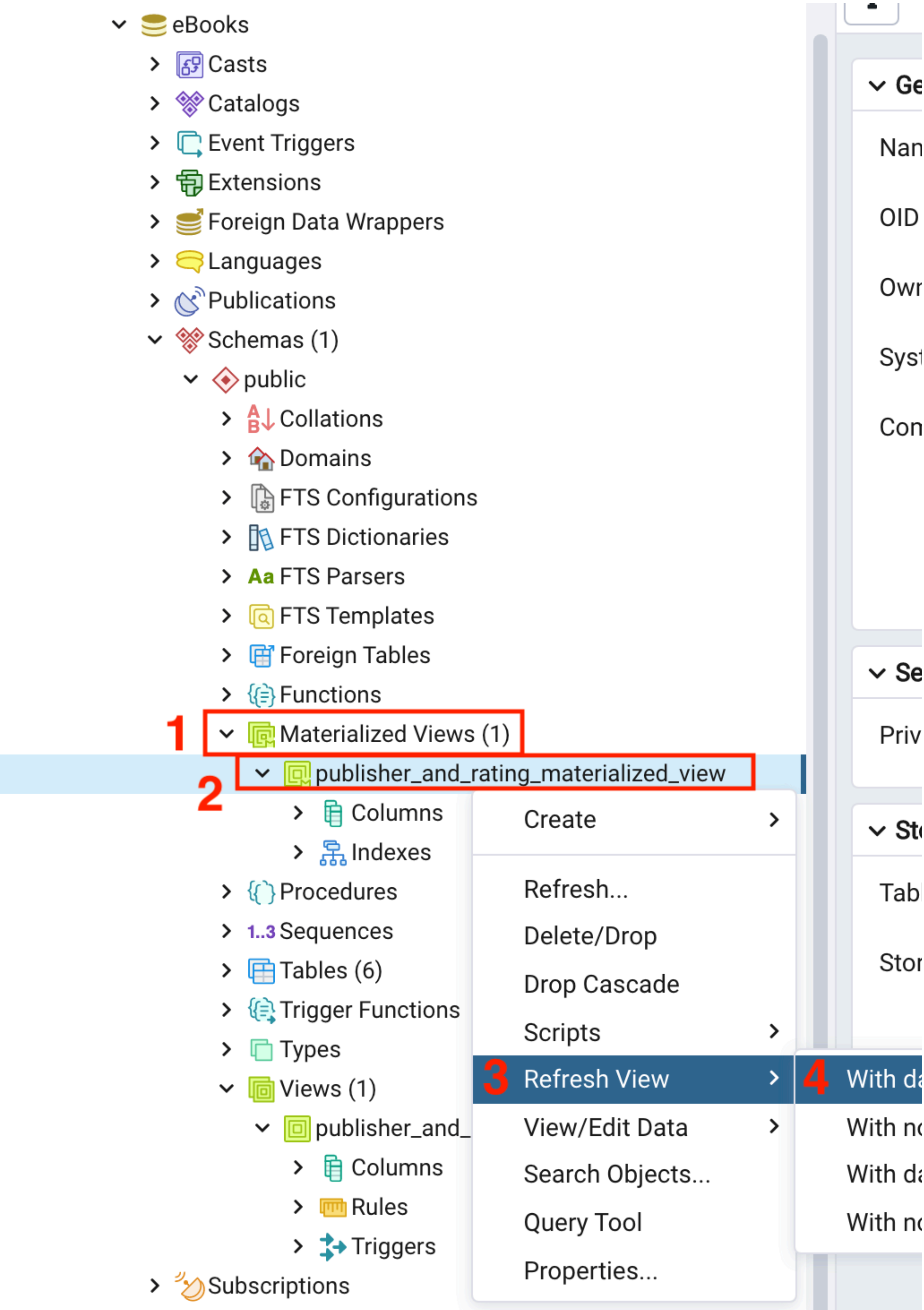
3. On the **code** tab, copy and paste the following code. Then click **Save**.

```
SELECT books.title, books.rating, publishers.name
FROM books INNER JOIN publishers ON books.publisher_id = publishers.publisher_id
```

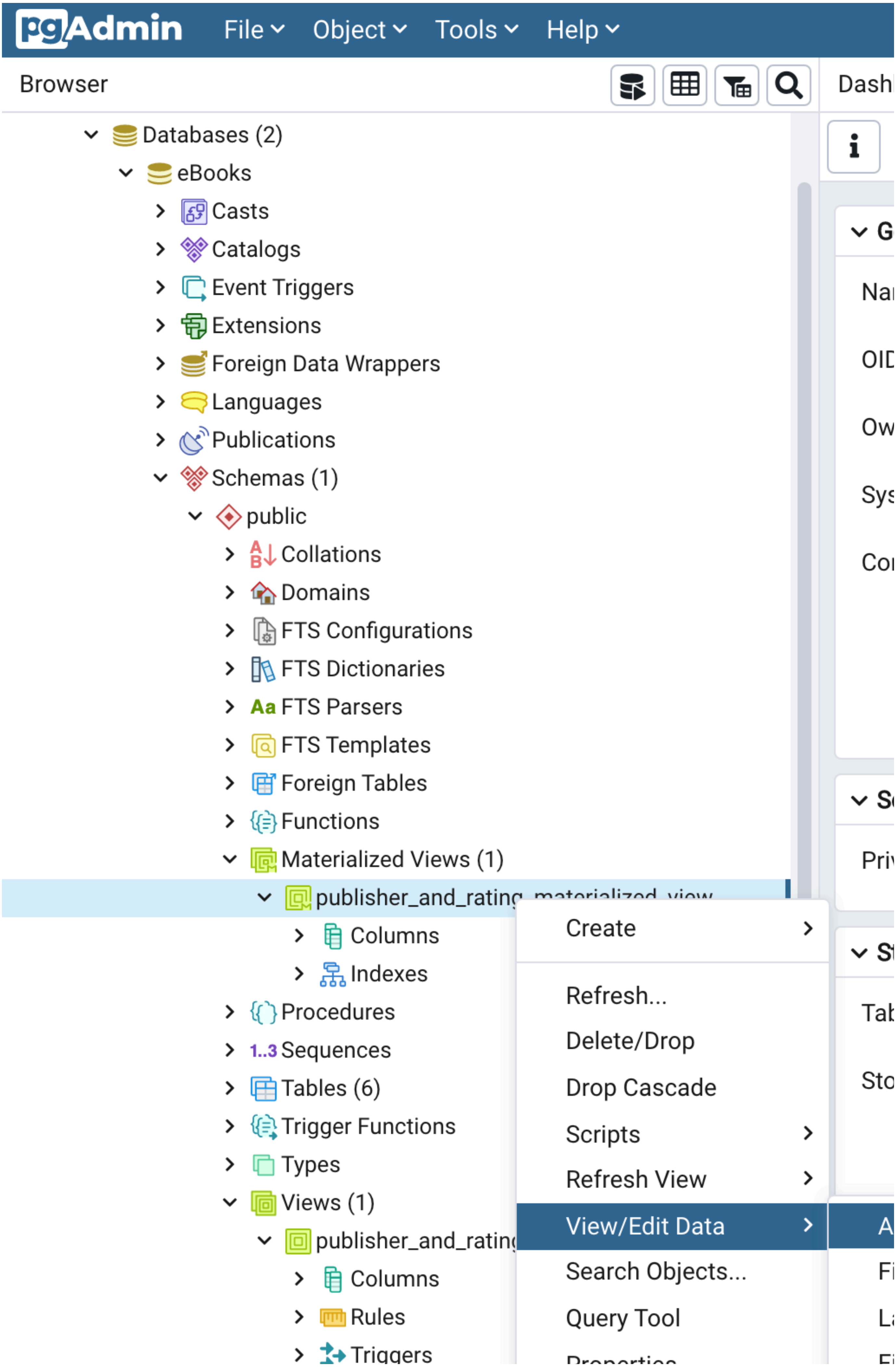


```
SELECT books.title, books.rating, publishers.name
FROM books INNER JOIN publishers ON books.publisher_id = publishers.publishe
```

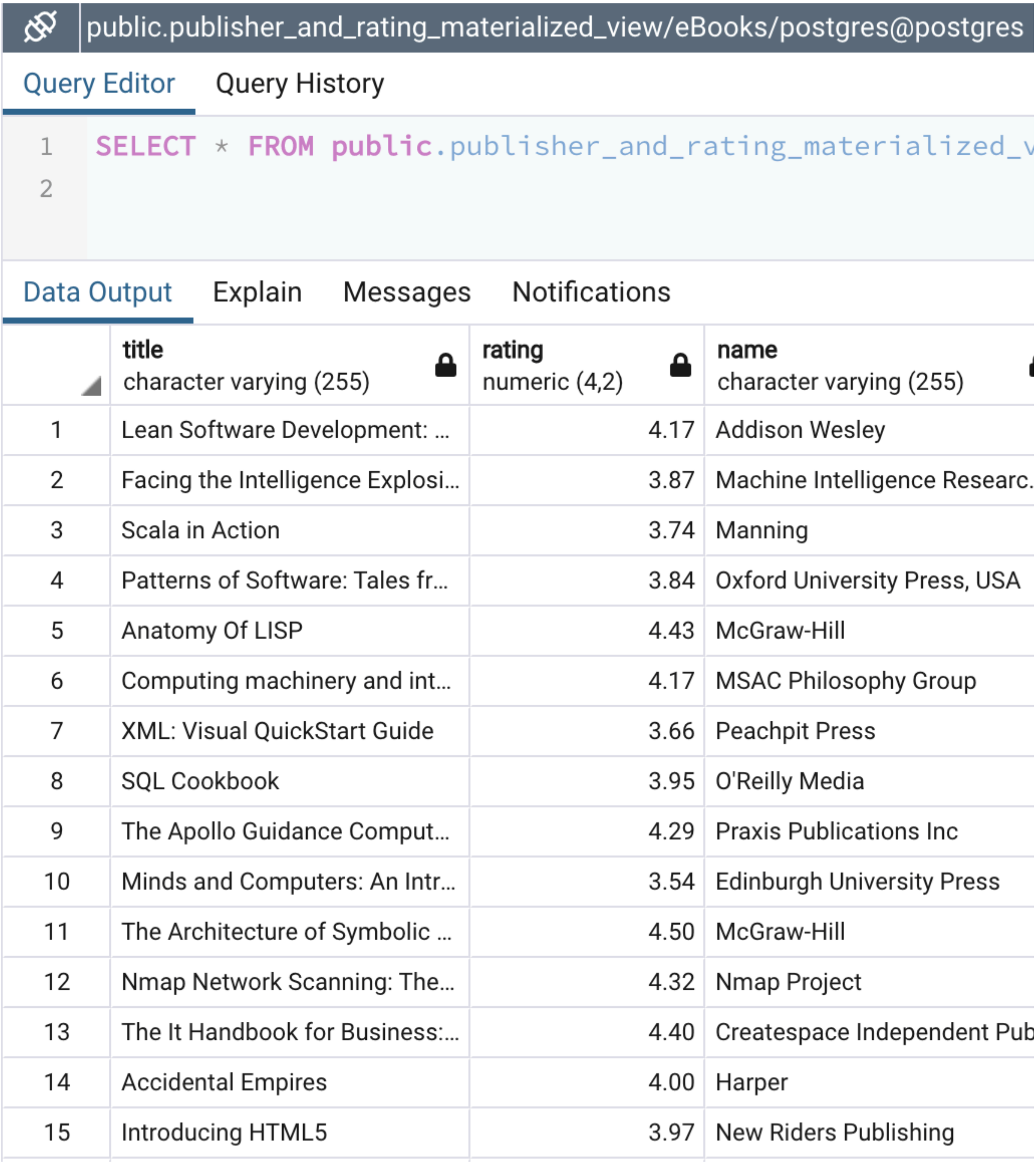
4. In the tree-view, expand **Materialized Views**. Right-click **publisher_and_rating_materialized_view** and go to **Refresh View > With data**.



5. Right-click **publisher_and_rating_materialized_view** again and go to **View/Edit Data** > **All Rows**.


pgAdmin
File
Object
Tools
Help
Browser
Dash
Databases (2)
eBooks
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas (1)
public
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views (1)
publisher_and_rating_materialized_view
Columns
Indexes
Procedures
Sequences
Tables (6)
Trigger Functions
Types
Views (1)
publisher_and_rating
Columns
Rules
Triggers
Create
Refresh...
Delete/Drop
Drop Cascade
Scripts
Refresh View
View/Edit Data
Search Objects...
Query Tool
Properties
A
Fi
L
Fi
G
Na
OID
Ow
Sys
Co
S
Pri
S
Tal
Sto

6. You will access the materialized view you created.


public.publisher_and_rating_materialized_view/eBooks/postgres@postgres
Query Editor
Query History
Data Output
Explain
Messages
Notifications

```
SELECT * FROM public.publisher_and_rating_materialized_v
```

	title character varying (255)	rating numeric (4,2)	name character varying (255)
1	Lean Software Development: ...	4.17	Addison Wesley
2	Facing the Intelligence Explosi...	3.87	Machine Intelligence Researc.
3	Scala in Action	3.74	Manning
4	Patterns of Software: Tales fr...	3.84	Oxford University Press, USA
5	Anatomy Of LISP	4.43	McGraw-Hill
6	Computing machinery and int...	4.17	MSAC Philosophy Group
7	XML: Visual QuickStart Guide	3.66	Peachpit Press
8	SQL Cookbook	3.95	O'Reilly Media
9	The Apollo Guidance Comput...	4.29	Praxis Publications Inc
10	Minds and Computers: An Intr...	3.54	Edinburgh University Press
11	The Architecture of Symbolic ...	4.50	McGraw-Hill
12	Nmap Network Scanning: The...	4.32	Nmap Project
13	The It Handbook for Business:...	4.40	Createspace Independent Pub
14	Accidental Empires	4.00	Harper
15	Introducing HTML5	3.97	New Riders Publishing

At first glance, it does not look too different from the regular view you created earlier in this lab. From the user perspective, it is essentially the same: you see the results of a query displayed in a table-like format. The difference is that this materialized view is cached in the database so someone can reaccess the data in the future without re-running the database query.

Conclusion

Congratulations! You have completed this lab and learned how to restore a database schema and data, create and execute a view, and create and execute a materialized view.

Author: Sandip Saha Joy

Other Contributors: David Pasternak

Skills Network