

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.

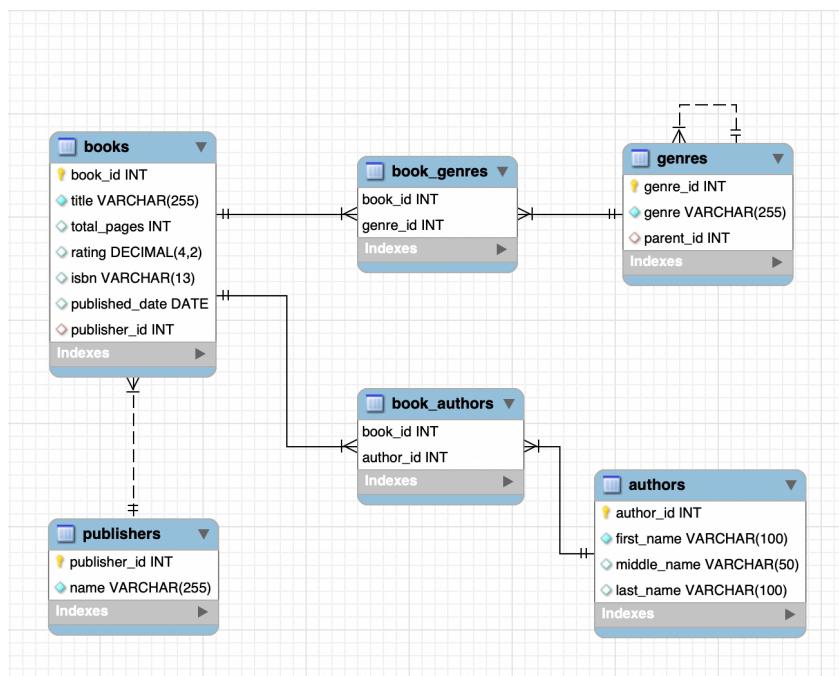


To complete this lab, you will utilize the PostgreSQL relational database service available as part of IBM Skills Network Labs (SN Labs) Cloud IDE. SN Labs is a virtual lab environment used in this course.

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.
 - o [eBooks_pgsql_dump.tar](#)
2. Click the **Skills Network** extension on the left side of the window.
3. Select the **DATABASES** menu and click **PostgreSQL**.
4. Click **Start**. PostgreSQL may take a few moments to start.

The screenshot shows the IBM Cloud IDE interface with the following steps highlighted:

- 1**: A red circle highlights the "PostgreSQL" icon in the sidebar.
- 2**: A red circle highlights the "PostgreSQL" service entry in the "DATABASES" section of the sidebar.
- 3**: A red circle highlights the "Start" button for the PostgreSQL service.

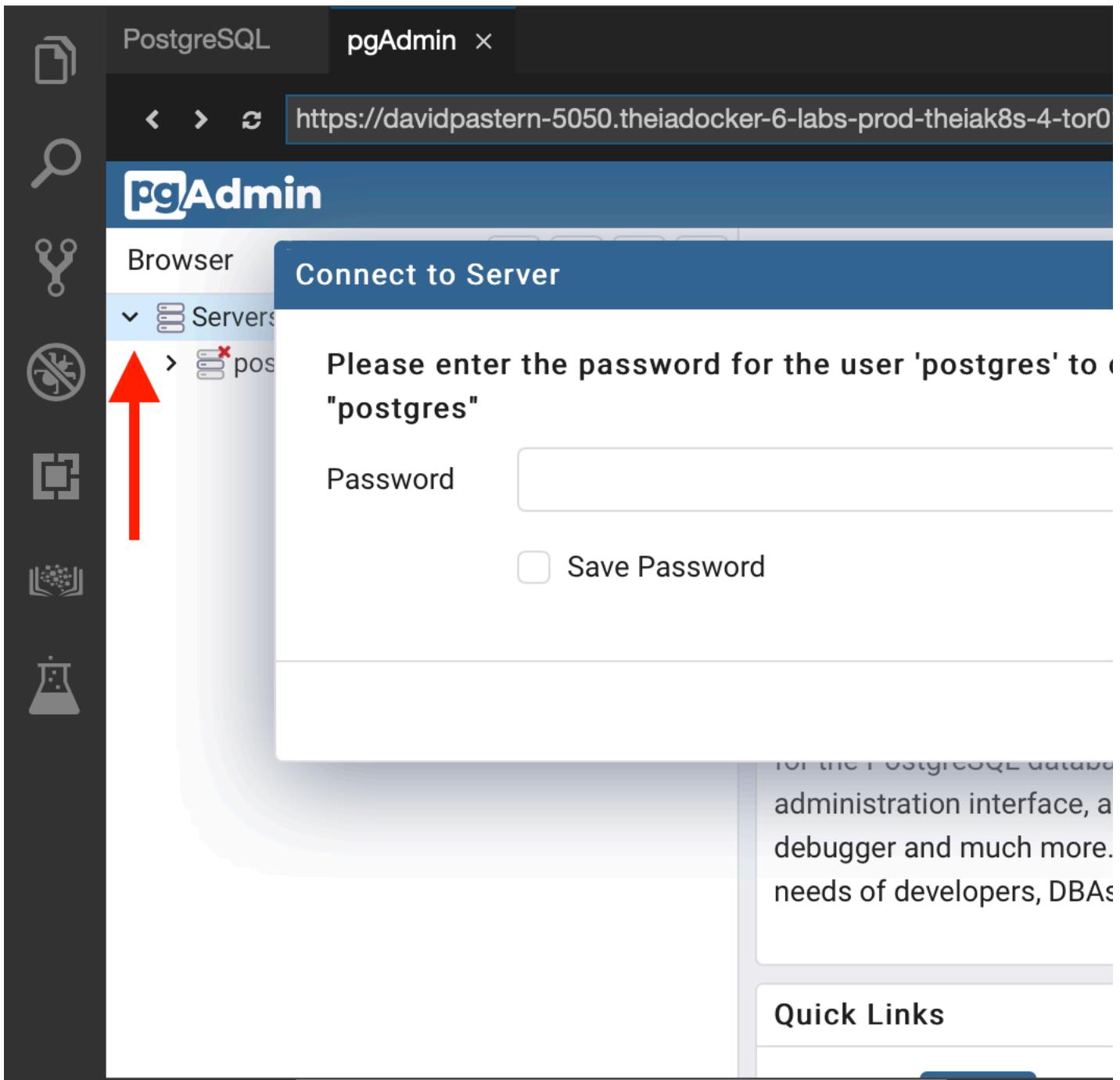
The main panel displays the PostgreSQL service details:

- PostgreSQL** (Status: INACTIVE)
- Version: v13.2 | Configuration: v5.0 | Environment: v13.2
- Connect to PostgreSQL and pgAdmin directly in your Skills
- Buttons: Start, Summary, Connection Information, Details
- Description: Get started with Postgres in a faster, easier way. To launch y

5. Open the pgAdmin graphical user interface (GUI) by selecting **pgAdmin** in the Cloud IDE interface.

The screenshot shows a dark-themed PostgreSQL setup interface. On the left, there's a vertical sidebar with icons for file operations, search, and database management. The main area has a title "PostgreSQL x" and a "Stop" button. Below it, three tabs are visible: "Summary" (underlined), "Connection Information", and "Details". A message states: "Your database and pgAdmin server are now ready to use and available with the on how to navigate PostgreSQL, please check out the Details section." Below this are fields for "Username" and "Password", each with a copy icon. A section titled "You can manage PostgreSQL via:" contains two buttons: "pgAdmin" (which is highlighted with a red box) and "New Terminal".

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



7. To retrieve your password, click **PostgreSQL** near the top of the interface.
8. Click **Copy** to the left of your password to copy the session password onto your clipboard.

The screenshot shows a dark-themed PostgreSQL management interface. On the left, there's a vertical sidebar with icons for file operations, search, and other database management tasks. The main area has a header with the title "PostgreSQL" and the status "ACTIVE". Below the title, it says "v13.2" for the database version, "v5.0" for the pgAdmin version, and "v13.2" for the connection port. A large red circle with the number "1" highlights the "Stop" button. Below the status bar, there are three tabs: "Summary" (underlined), "Connection Information", and "Details". The "Summary" tab contains a message: "Your database and pgAdmin server are now ready to use and available with the following connection details. For more information on how to navigate PostgreSQL, please check out the Details section." It also shows fields for "Username" and "Password", each with a copy icon. A red circle with the number "2" highlights the "Copy" icon next to the password field. Below these fields, it says "You can manage PostgreSQL via:" followed by two buttons: "pgAdmin" and a terminal icon. At the bottom, it says "Or to interact with the database in the terminal, select one of these options:" followed by two buttons: "PostgreSQL CLI" and "New Terminal".

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.

← → ⌂ ⌂ sandipsahajo-5050.theiadocker-27.proxy.cognitivecl

pgAdmin File ▾ Object ▾ Tools ▾ Help ▾

Browser     Dashboard Properties SQL

>  Servers

Welcome

 pgAdn
Management

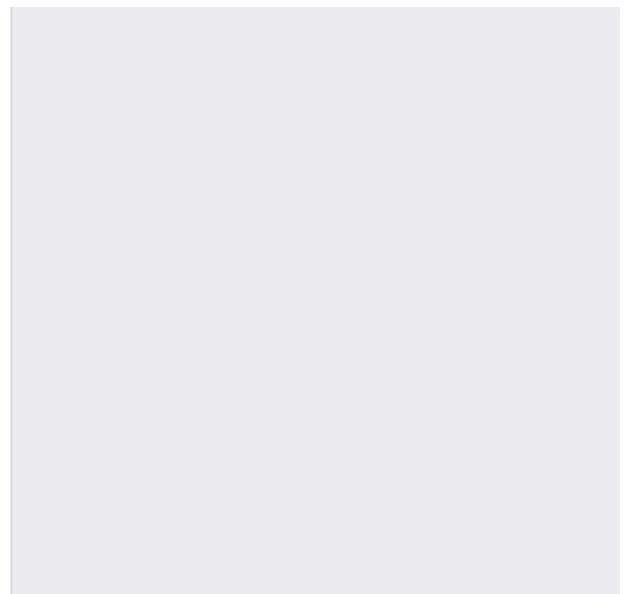
Feature rich | Maximise

pgAdmin is an Open Source admir
is designed to answer the needs o

Quick Links

Getting Started

 PostgreSQL Document



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**.

The screenshot shows the pgAdmin 4 interface. The top navigation bar includes 'pgAdmin' and dropdown menus for 'File', 'Object', 'Tools', and 'Help'. Below the navigation is a toolbar with icons for 'Browser', 'Dashboard', 'Properties', and 'SQL'. The main area is titled 'Server sessions' and shows a list of sessions with a count of 7.

The 'Browser' panel on the left displays a tree structure:

- Servers (1)** (highlighted with a red box, step 1)
- postgres** (highlighted with a red box, step 2)
- Databases (1)** (highlighted with a red box, step 3)
- postgres** (under Databases)
- Sub-items under 'postgres':
 - Casts
 - Catalog
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas
 - Subscriptions
- Login/Group Roles
- Tablespaces

A context menu is open over the 'Databases (1)' node, with the 'Create' option highlighted. A mouse cursor is pointing at the 'Database...' button in the 'Create' submenu.

Create - Database

General Definition Security Parameters Advanced SQL

Database

eBooks

Owner

 postgres

Comment



 Cancel

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

pgAdmin File ▾ Object ▾ Tools ▾ Help ▾

Browser

Servers (1) postgres

Databases (2) eBooks

Restore...

Refresh... Delete/Drop CREATE Script Disconnect Database... Generate ERD (Beta) Maintenance... Backup... Properties... Query Tool Publications Schemas Subscriptions

Database s

1

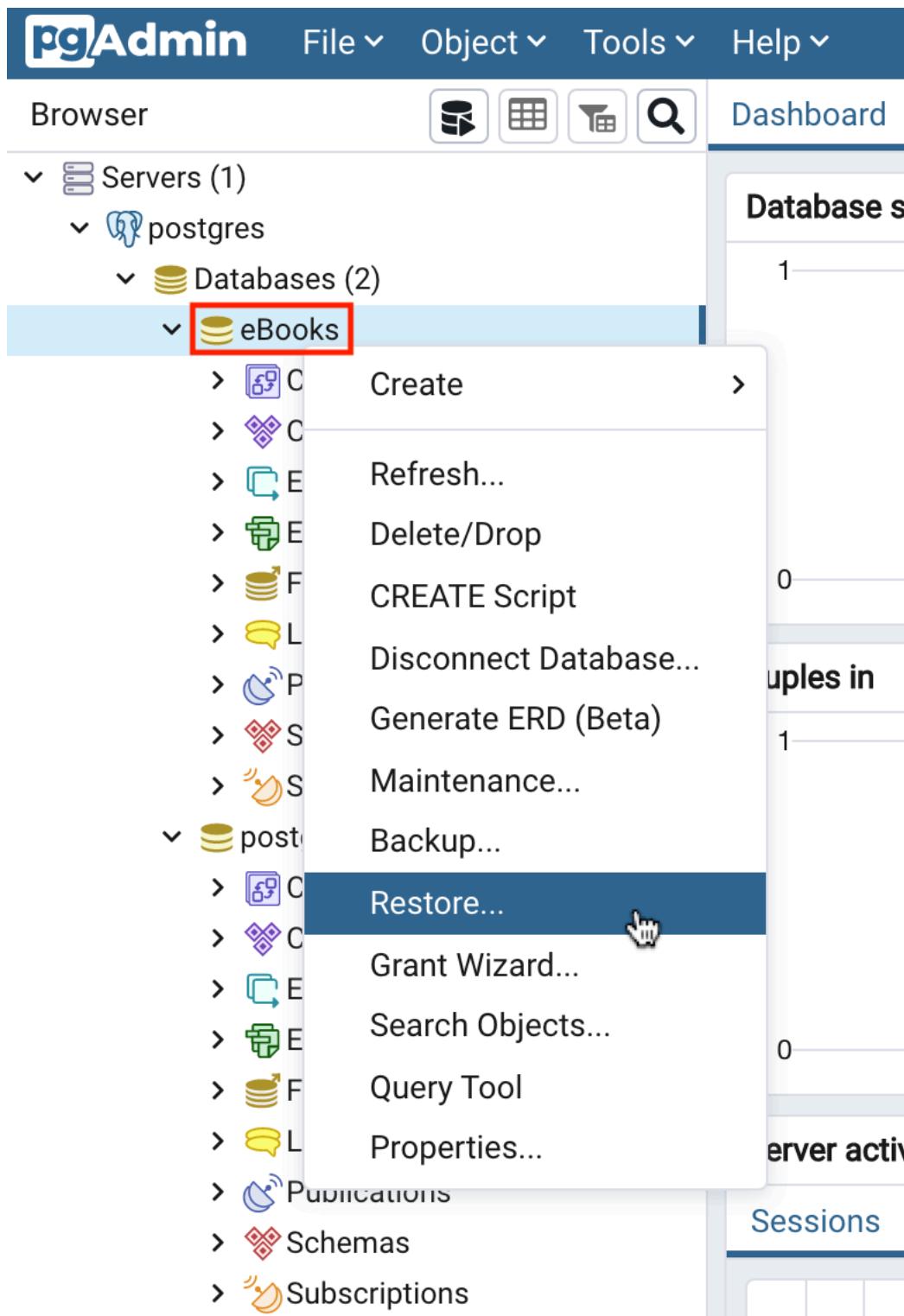
uples in

0

1

server activ

Sessions



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

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

Restore (Database: eBooks)

General

Restore options

Format

Custom or tar

Filename

Number of jobs

Role name

Select an item...



- o Click **Upload File**.

Select file

/var/lib/pgadmin/

Name	Size
sessions	4.0 kB
storage	4.0 kB

Show hidden files and folders?

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

Select file

/var/lib/pgadmin/

Double click on this space

Drop files here to upload. The file size limit (per file) is 50

Show hidden files and folders?

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

Select file



/var/lib/pgadmin/



76 KB

eBooks_pgsql_d...

100%

Drop files here to upload. The file size limit (per file) is 50

Show hidden files and folders?

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

Select file

/var/lib/pgadmin/eBooks_pgsql_dump.tar

Name	Size
eBooks_pgsql_dump.tar	74.2 kB
pgadmin4.db	156.0 kB
sessions	4.0 kB
storage	4.0 kB

Show hidden files and folders?

- Now switch to the **Restore options** tab.

Restore (Database: eBooks)

General **Restore options** 

Format

Custom or tar

Filename

/var/lib/pgadmin/eBooks_pgsql_dump.tar

Number of jobs

Role name

Select an item...



- Under **Disable**, set the **Trigger** option to **Yes**. Then click **Restore**.

Restore (Database: eBooks)

General Restore options

Queries

Include CREATE DATABASE statement

Clean before restore

Single transaction

Disable

Trigger

No data for Failed Tables

Help

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)

1 postres **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

5 Create > 6 View...

Refresh...

- > Casts
- > Catalogs
- > Event Triggers
- > Extensions
- > Foreign Data Wrappers
- > Languages

Grant Wizard...
Search Objects...
Query Tool

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

Create - View

General Definition Code Security SQL

Name	publisher_and_rating_view
Owner	postgres
Schema	public
Comment	

Buttons: Cancel Reset Save

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

2. 2
1. SELECT books.title, books.rating, publishers.name
2. FROM books INNER JOIN publishers ON books.publisher_id = publishers.publisher_id

Copied!

Create - View

General Definition **Code** Security SQL

```
1 SELECT books.title, books.rating, publishers.name  
2 FROM books INNER JOIN publishers ON books.publisher_id = pu  
3
```



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 Properties

Servers (1) postres 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 Views (1) publisher_and_rating_view Columns Rules Create

Database sessions
1

Tuples in
18
16
14
12
10
8
6
4
2
0

Server activity
Sessions Locks

PID
83

1 2

The screenshot shows a left-hand sidebar with a tree view of database objects under the 'about:blank' connection. The tree includes categories like Triggers, Subscriptions, and various database-related components. A context menu is open on the right side, listing actions such as Refresh..., Delete/Drop, Drop Cascade, Scripts, View/Edit Data, Search Objects..., Query Tool, and Properties... . The 'View/Edit Data' option is highlighted with a red border and a red number '3' to its left. To the right of the menu, there are four numbered buttons: 'All Rows' (red '4'), 'First 100', 'Last 100', and 'Filtered'.

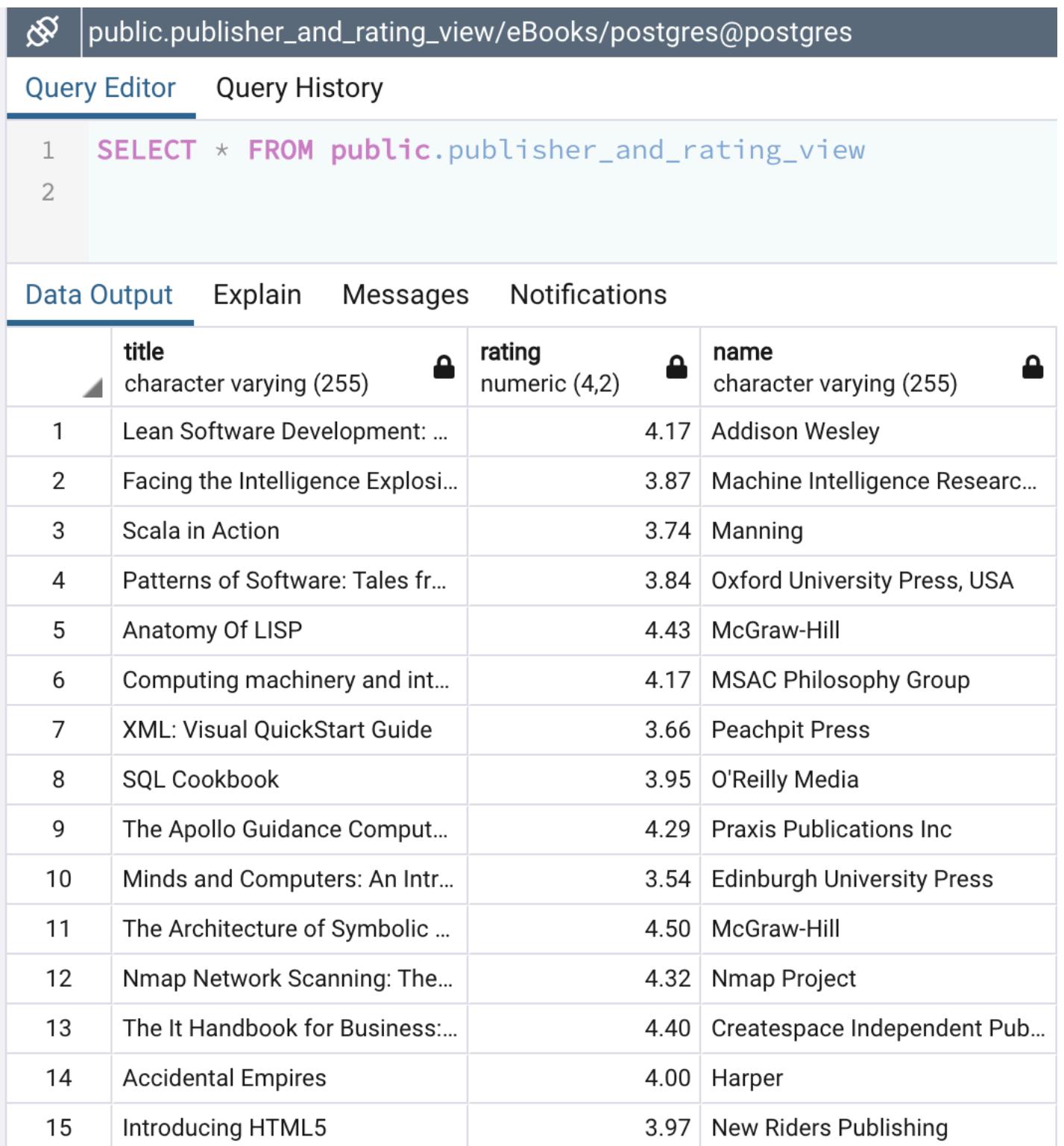
- > Triggers
- > Subscriptions
- < postgres
 - > Casts
 - > Catalogs
 - > Event Triggers
 - > Extensions
 - > Foreign Data Wrappers
 - > Languages
 - < Publications

about:blank

- Refresh...
- Delete/Drop
- Drop Cascade
- Scripts >
- 3 View/Edit Data > 4 All Rows**
- Search Objects...
- Query Tool
- Properties...

First 100
Last 100
Filtered

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



The screenshot shows a PostgreSQL query editor interface. At the top, there's a header bar with a dollar sign icon and the text "public.publisher_and_rating_view/eBooks/postgres@postgres". Below the header, there are tabs for "Query Editor" (which is selected) and "Query History". The main area contains a numbered query and its results.

```

1  SELECT * FROM public.publisher_and_rating_view
2

```

Below the query results, there are tabs for "Data Output" (selected), "Explain", "Messages", and "Notifications". The "Data Output" tab displays a table with 15 rows of data from the materialized 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

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

pgAdmin File ▾ Object ▾ Tools ▾ Help ▾

Browser

Servers (1) Dashboard Properties

1. Servers (1)
 2. postres
 3. Databases (2)
 4. eBooks
 5. Casts
 6. Catalogs
 7. Event Triggers
 8. Extensions
 9. Foreign Data Wrappers
 10. Languages
 11. Publications
 12. Schemas (1)
 13. public
 14. Collations
 15. Domains
 16. FTS Configurations
 17. FTS Dictionaries
 18. FTS Parsers
 19. FTS Templates
 20. Foreign Tables
 21. Functions
 22. Materialized Views
 23. Procedures
 24. Sequences
 25. Tables (6)
 26. Trigger Functions
 27. Types
 28. Views (1)
 29. Subscriptions

Materialized Views Create Refresh... Grant Wizard... Search Objects... Query Tool

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

 Create - Materialized View**General**

Definition

Storage

Parameter

Security

SQL

Name

publisher_and_rating_materialized_view

Owner

 postgres

Schema

 public

Comment

 Cancel

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

```
1. 1
2. 2
1. SELECT books.title, books.rating, publishers.name
2. FROM books INNER JOIN publishers ON books.publisher_id = publishers.publisher_id
```

Copied!

Create - Materialized View

General **Definition** Storage Parameter Security SQL

```
1 SELECT books.title, books.rating, publishers.name  
2 FROM books INNER JOIN publishers ON books.publisher_id = pu  
3
```



✖ Cancel

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

1 **Materialized Views (1)**

2 **publisher_and_rating_materialized_view**

3 Refresh View

4 With data

View/Edit Data

Search Objects...

Query Tool

Properties...

The screenshot shows a database management interface with a sidebar and a main content area. The sidebar contains sections for 'Gene', 'Name', 'OID', 'Owner', 'System', 'Comm', 'Secu', 'Privilec', 'Stora', 'Tables', and 'Storage'. The main content area displays a tree view of database objects under 'eBooks'. The 'Materialized Views' node is selected, highlighted with a red border and numbered 1. A context menu is open over the 'publisher_and_rating_materialized_view' node, also highlighted with a red border and numbered 2. The menu items are: Create, Refresh..., Delete/Drop, Drop Cascade, Scripts, Refresh View (selected and numbered 3), and With data (selected and numbered 4). Other menu items include View/Edit Data, Search Objects..., Query Tool, and Properties... . The 'Refresh View' and 'With data' items are also highlighted with red borders.

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

Browser



Dashboard

- ▼ 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_view
 - > Columns

publisher_and_rating_materialized_view

- Create >
- Refresh...
- Delete/Drop
- Drop Cascade
- Scripts >
- Refresh View >
- View/Edit Data >**
- All R
- Search Objects...
- First

- > Columns
- > Rules
- > Triggers

> Subscriptions

Query Tool

Properties...

Last

Filte

6. You will access the materialized view you created.

public.publisher_and_rating_materialized_view/eBooks/postgres@postgres

Query Editor Query History

```
1  SELECT * FROM public.publisher_and_rating_materialized_view
```

Data Output Explain Messages Notifications

	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