



Hands-on Lab : Views in PostgreSQL

Estimated time needed: 15 minutes

In this lab, you will learn how to create and execute views and materialized views in the PostgreSQL database service using the pgAdmin graphical user interface (GUI) tool. Materialized views behave differently compared to regular views. In materialized views, the result set is materialized, or saved for future use. You can't insert, update, or delete rows like in regular views. Essentially, materialized views store the results of a database query as a separate table-like object so that the query results can be accessed at a later time 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 efficiently store, manipulate, and retrieve the data.

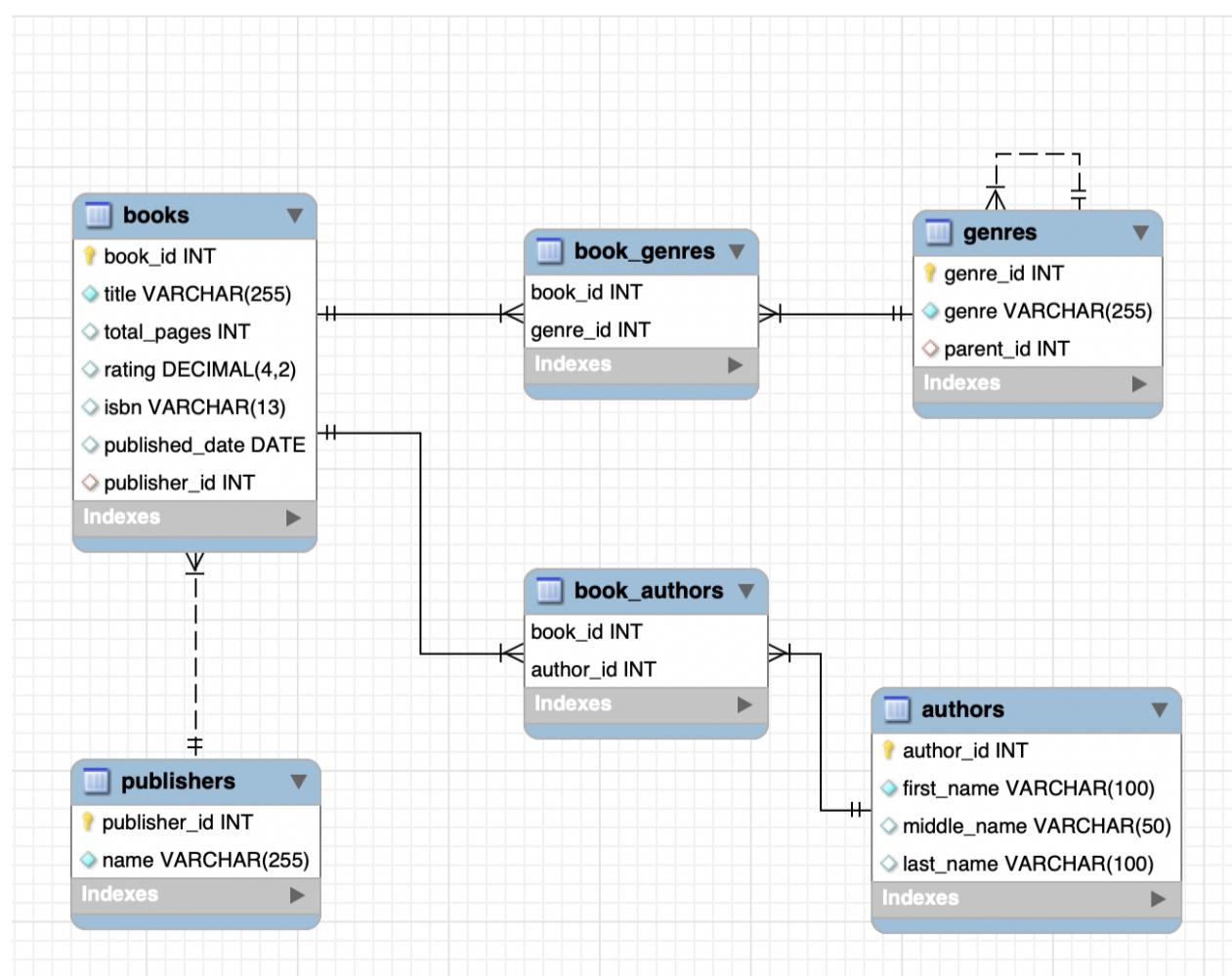


To complete this lab you will utilize the PostgreSQL relational database service available as part of IBM Skills Network Labs (SN Labs) Cloud IDE. SN Labs is a virtual lab environment used in this course.

Database Used in this Lab

The eBooks database has been used in this 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.

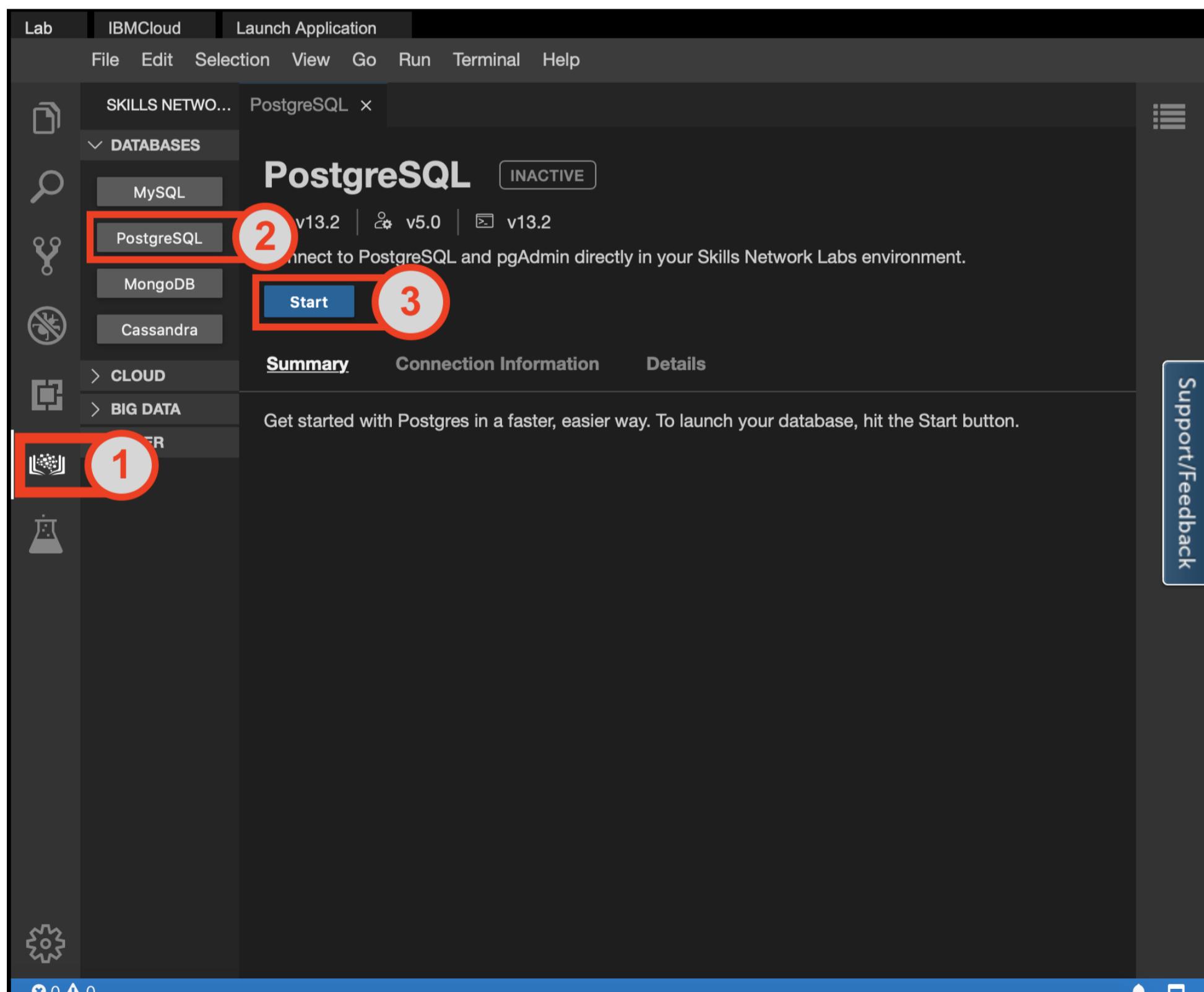
Lab Structure

In this exercise, you will go through three tasks where you will 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 **eBooks** PostgreSQL dump file (containing the eBooks database schema and data) below to your local computer storage.
 - o [eBooks_pgsql_dump.tar](#)
2. Click on the Skills Network extension button on the left side of the window.
3. Open the "DATABASES" drop down menu and click on "PostgreSQL"
4. Click on the "Start" button. PostgreSQL may take a few moments to start.



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

The screenshot shows the PostgreSQL setup interface. At the top, there are icons for file operations (New, Open, Save, Delete, Copy, Paste, Find, Help). A blue button labeled "Stop" is visible. Below it, three tabs are shown: "Summary" (selected), "Connection Information", and "Details". A message states: "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." Below this, fields for "Username" and "Password" are displayed, each with a copy icon. A section titled "You can manage PostgreSQL via:" contains a "pgAdmin" button, which is highlighted with a red box and a copy icon. Another section says "Or to interact with the database in the terminal, select one of these options:" followed by "PostgreSQL CLI" and "New Terminal" buttons.

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

The screenshot shows the pgAdmin interface. The title bar says "PostgreSQL" and "pgAdmin". The address bar shows the URL: "https://davidpastern-5050.theiadocker-6-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai/browser/". On the left, a sidebar has icons for Browser, Servers, and a connection to "postgres". A red arrow points from this icon towards the center. The main area is a "Connect to Server" dialog box with the title "Connect to Server". It displays the message: "Please enter the password for the user 'postgres' to connect the server - 'postgres'". Below this is a "Password" input field and a "Save Password" checkbox. At the bottom are "Cancel" and "OK" buttons. To the right of the dialog, there is descriptive text about pgAdmin and a "Quick Links" section.

7. To retrieve your password, click on the "PostgreSQL" tab near the top of the interface.

8. Click on the Copy icon to the left of your password to copy the session password onto your clipboard.

PostgreSQL x pgAdmin

PostgreSQL

ACTIVE

v13.2 | v5.0 | v13.2

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

Stop

Summary **Connection Information** **Details**

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.

Username: [REDACTED]

Password: [REDACTED] **2**

You can manage PostgreSQL via:

pgAdmin

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

PostgreSQL CLI **New Terminal**

9. Navigate back to the "pgAdmin" tab and paste in your password, then click **OK**

10. You will then be able to access the pgAdmin GUI tool.

sandipsahajo-5050.theiadocker-27.proxy.cognitiveclass.ai/browser/

pgAdmin File Object Tools Help

Browser Servers

Welcome

pgAdmin
Management Tools for PostgreSQL
Feature rich | Maximises PostgreSQL | Open Source

pgAdmin is an Open Source administration and management tool for the PostgreSQL database. It includes a graphical administration interface, an SQL query tool, a procedural code debugger and much more. The tool is designed to answer the needs of developers, DBAs and system administrators alike.

Quick Links

Add New Server Configure pgAdmin

Getting Started

PostgreSQL Documentation pgAdmin Website Planet PostgreSQL Community Support

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 name of the database and click **Save**.

The screenshot shows the pgAdmin interface. The top navigation bar includes 'File', 'Object', 'Tools', and 'Help'. Below the navigation is a toolbar with icons for 'Dashboard', 'Properties', 'SQL', and 'Statistics'. The main area has tabs for 'Browser' (highlighted with a red box and number 1), 'Dashboard' (selected and highlighted with a blue bar), 'Properties', 'SQL', and 'Statistics'. In the 'Browser' tab, the tree view shows 'Servers (1)' (highlighted with a red box and number 2), which contains 'postgres' (highlighted with a red box and number 3), and 'Databases (1)'. Under 'postgres', there are several items: 'Create' (selected and highlighted with a blue bar), 'Refresh...', and a list of objects: 'Casts', 'Catal', 'Event Triggers', 'Extensions', 'Foreign Data Wrappers', 'Languages', 'Publications', 'Schemas', 'Subscriptions', 'Login/Group Roles', and 'Tablespaces'. To the right of the tree view is a 'Server sessions' panel showing a count of 7. Below it is a 'Tuples in' panel showing a count of 1. A context menu is open over the 'Create' item, with a cursor pointing at the 'Database...' option. A modal window titled 'Create - Database' is open in the foreground. It has tabs for 'General' (selected and highlighted with a blue bar), 'Definition', 'Security', 'Parameters', 'Advanced', and 'SQL'. The 'General' tab contains fields for 'Database' (set to 'eBooks') and 'Owner' (set to 'postgres'). The 'Comment' field is empty. At the bottom of the modal are buttons for 'Cancel', 'Reset', and 'Save' (highlighted with a red box).

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

The screenshot shows the pgAdmin interface. In the top navigation bar, the 'File', 'Object', 'Tools', and 'Help' menus are visible. Below the navigation bar, there are four icons: a server, a grid, a funnel, and a magnifying glass. The 'Dashboard' tab is selected. On the left, the 'Browser' panel shows a tree structure: 'Servers (1) > postgres > Databases (2) > eBooks'. The 'eBooks' node is highlighted with a red box. A context menu is open over the 'eBooks' node, listing the following options: 'Create', 'Refresh...', 'Delete/Drop', 'CREATE Script', 'Disconnect Database...', 'Generate ERD (Beta)', 'Maintenance...', 'Backup...', 'Restore...', 'Grant Wizard...', 'Search Objects...', 'Query Tool', 'Properties...', 'Publications', 'Schemas', and 'Subscriptions'. The 'Restore...' option is highlighted with a blue background and a cursor icon pointing at it.

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

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

Restore (Database: eBooks)

General Restore options

Format	Custom or tar
Filename	<input type="text"/> ...
Number of jobs	<input type="text"/>
Role name	Select an item...

Information:

Actions: Cancel Restore

- Click the **Upload File** button.

Select file

/var/lib/pgadmin/

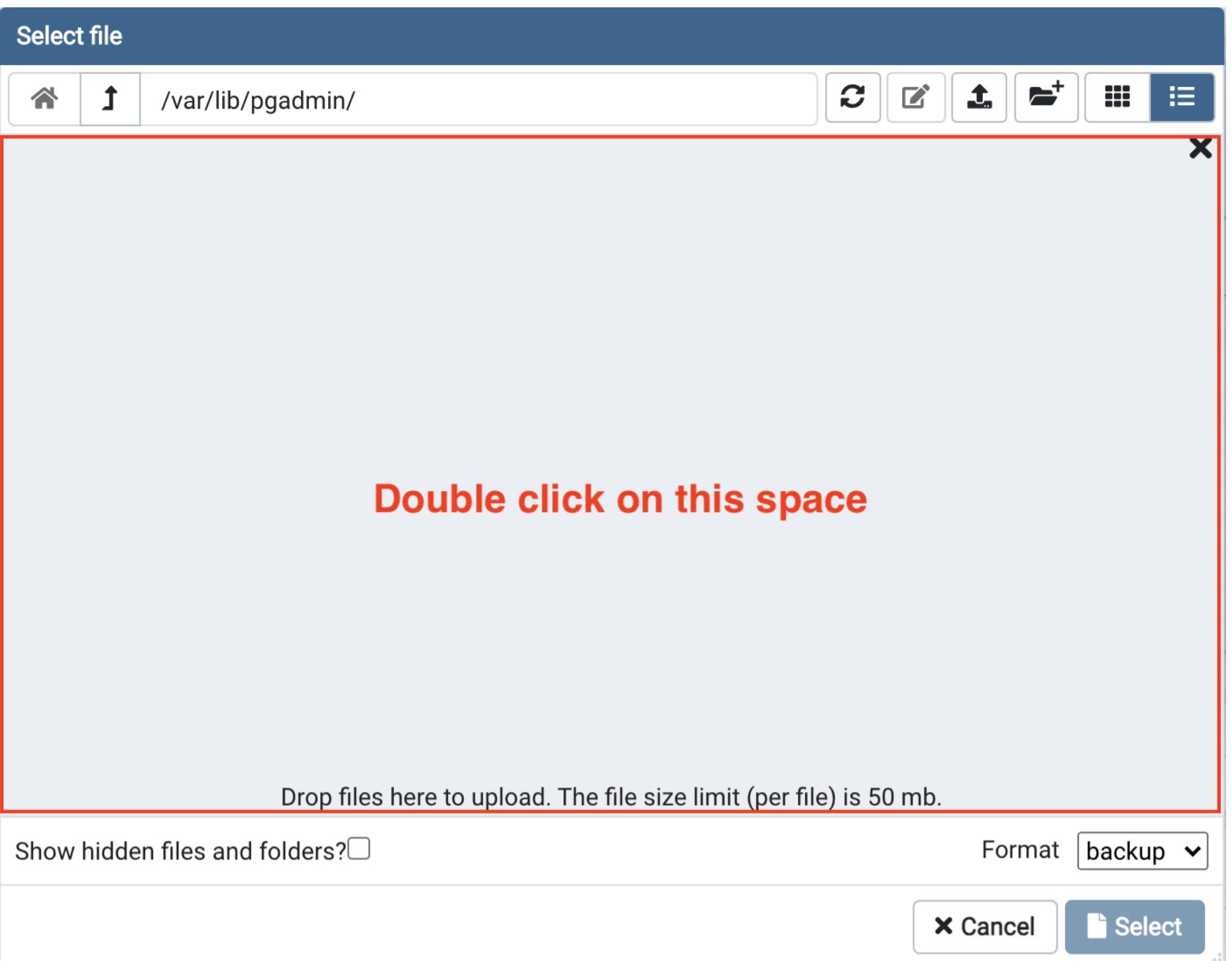
Name	Size	Modified
sessions	4.0 kB	Mon Mar 29 10:20:20 2021
storage	4.0 kB	Mon Mar 29 10:04:10 2021

Show hidden files and folders?

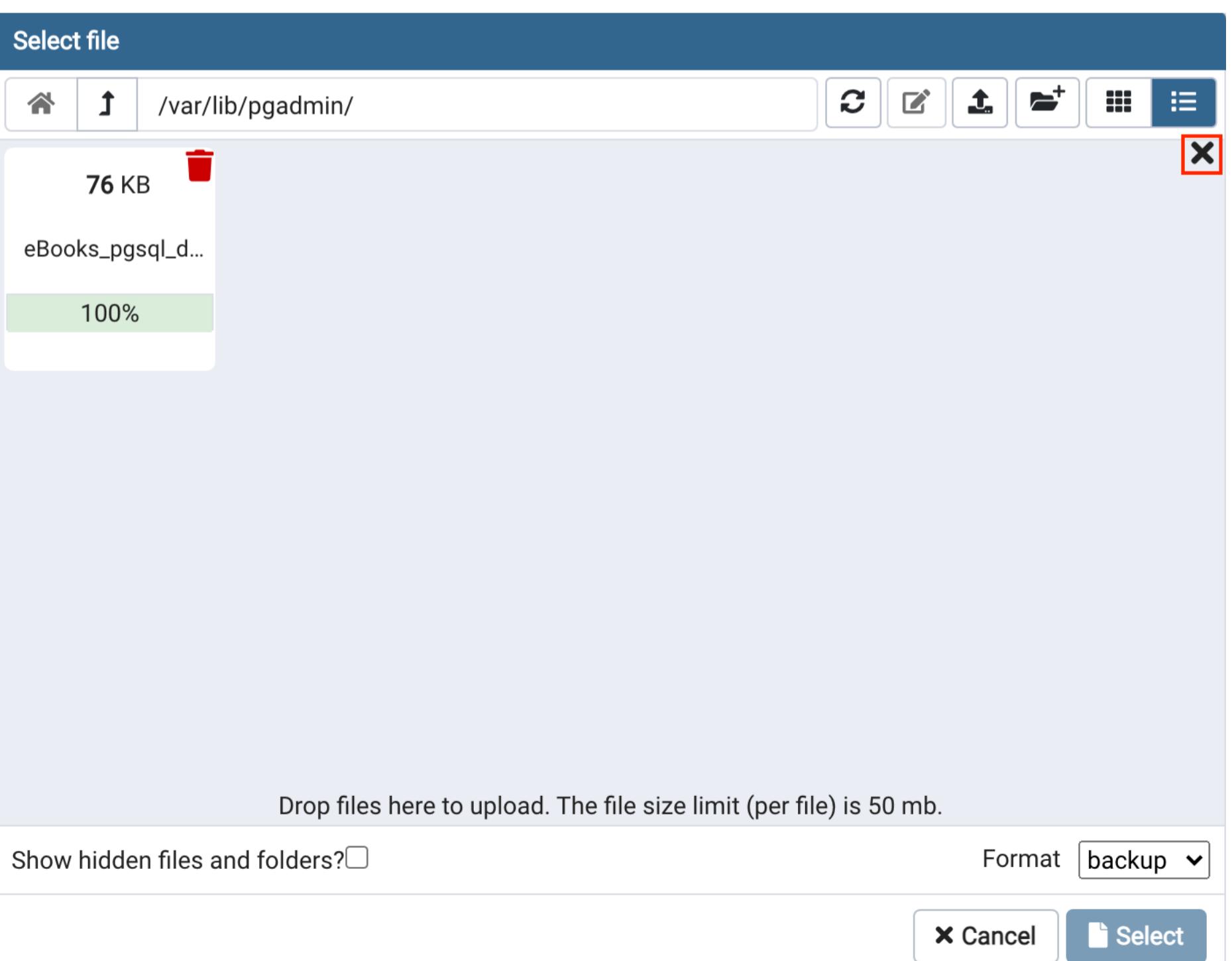
Format backup

Actions: Cancel Select

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



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



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

Select file

/var/lib/pgadmin/eBooks_pgsql_dump.tar

Name	Size	Modified
eBooks_pgsql_dump.tar	74.2 kB	Mon Mar 29 10:18:23 2021
pgadmin4.db	156.0 kB	Mon Mar 29 10:15:25 2021
sessions	4.0 kB	Mon Mar 29 10:04:48 2021
storage	4.0 kB	Mon Mar 29 10:04:10 2021

Show hidden files and folders?

Format **All Files**

Select

- Now switch to **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...

Cancel **Restore**

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

Restore (Database: eBooks)

General Restore options

Queries

Include CREATE DATABASE statement No

Clean before restore No

Single transaction No

Disable Trigger Yes

No data for Failed Tables No

Restore

Cancel

i **?**



Task B: Create and execute a view

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

The screenshot shows the pgAdmin 4 interface with the following details:

- Top Bar:** File ▾, Object ▾, Tools ▾, Help ▾.
- Toolbar:** Browser, Dashboard, Properties (selected), SQL, Statistics, Dependencies.
- Servers:** Servers (1) > postgres > Databases (2).
- Database Structure:**
 - 1** eBooks > Casts, Catalogs, Event Triggers, Extensions, Foreign Data Wrappers, Languages, Publications.
 - 2** Schemas (1) > public.
 - 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.
- Context Menu (Open at Step 4):**
 - 5** Create > Refresh..., Grant Wizard..., Search Objects..., Query Tool.
 - 6** View... (highlighted with a mouse cursor).

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

Create - View

X

General Definition Code Security SQL

Name: publisher_and_rating_view

Owner: postgres

Schema: public

Comment:

i ? Cancel Reset Save

- On the **Code** tab, copy and paste the code below. Then click **Save**.

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



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 = publishers.publisher_id  
3
```



Cancel

Reset

Save

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

pgAdmin File Object Tools Help

Browser Dashboard Properties SQL Statistics

Servers (1)

1 Servers (1)

2 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

Views (1)

publisher_and_rating_view

Columns

Rules

Triggers

Subscriptions

postgres

Casts

Catalogs

Event Triggers

Extensions

Foreign Data Wrappers

Languages

Publications

Dashboard

Properties

SQL

Statistics

Database sessions

1

0

Tuples in

18
16
14
12
10
8
6
4
2
0

Server activity

Sessions Locks Prepared Transactions

	PID	User	A
	83	postgres	p

1 **Views (1)**

2 **publisher_and_rating_view**

3 **View/Edit Data**

4 **All Rows**

Create >

Refresh...

Delete/Drop

Drop Cascade

Scripts >

Search Objects...

Query Tool

Properties...

First 100 Rows

Last 100 Rows

Filtered Rows...

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



public.publisher_and_rating_view/eBooks/postgres@postgres

Query Editor

Query History

```

1  SELECT * FROM public.publisher_and_rating_view
2

```

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

Task C: Create and execute a materialized view

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

The screenshot shows the pgAdmin interface with the following steps highlighted:

1. In the left sidebar under 'eBooks' database, 'Materialized Views' is selected.
2. In the left sidebar under 'public' schema, 'Materialized Views' is selected.
3. In the left sidebar under 'public' schema, 'Functions' is selected.
4. In the left sidebar under 'public' schema, 'Materialized Views' is selected.
5. A context menu is open with the following options:
 - Create
 - Refresh...
 - Grant Wizard...
 - Search Objects...
 - Query Tool
6. The 'Create' option in the context menu is selected.

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

Owner ▼

Schema ▼

Comment

i ? ✖ Cancel ↻ Reset Save

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

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

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 = publishers.publisher_id
3
```

i **?** **Cancel** **Reset** **Save**

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

The screenshot shows the PostgreSQL Management Studio interface. On the left, a navigation tree displays various database objects under 'eBooks' and 'Schemas (1)'. Under 'Schemas (1)', there is a 'public' schema which contains 'Materialized Views (1)' and other objects like 'Functions'. A specific materialized view named 'publisher_and_rating_materialized_view' is selected and highlighted with a red box. A context menu is open for this object, with step numbers 1 through 4 indicating a sequence of actions:

- 1** Materialized Views (1)
- 2** publisher_and_rating_materialized_view
- 3** Refresh View
- 4** With data (highlighted)

The context menu options include:

- Create
- Refresh...
- Delete/Drop
- Drop Cascade
- Scripts
- Refresh View
- View/Edit Data
- Search Objects...
- Query Tool
- Properties...

On the right, a properties panel is visible with sections for General, Security, and Storage. The 'General' section includes fields for Name, OID, Owner, System materialized view?, and Comment.

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

The screenshot shows the pgAdmin interface with the following details:

- Top Bar:** pgAdmin, File ▾, Object ▾, Tools ▾, Help ▾
- Toolbar:** Browser, Databases, eBooks, Casts, Catalogs, Event Triggers, Extensions, Foreign Data Wrappers, Languages, Publications, Schemas, Subscriptions.
- Dashboard:** Dashboard, Properties (selected), SQL.
- Properties Panel:** General, Security, Storage.
- Object List:** 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 (selected).
- Context Menu (Open at publisher_and_rating):** Create, Refresh..., Delete/Drop, Drop Cascade, Scripts, Refresh View, View/Edit Data (selected), Search Objects..., Query Tool, Properties....
- View/Edit Data Submenu (Selected):** All Rows (selected), First 100 Rows, Last 100 Rows, Filtered Rows... (with a hand cursor icon).

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
2
```

[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

As you can see, at first glance it doesn't look too different from the regular view you created earlier in this lab - indeed, from the user perspective it's 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 that the data can be accessed again at a future time without having to re-run the database query, which can be intensive on the server depending on the complexity of the query and the size of the table being queried.

Congratulations! You have completed this lab, and you are ready for the next topic.

Author

- [Sandip Saha Joy](#)

Other Contributors

- [David Pasternak](#)

Changelog

Date	Version	Changed by	Change Description
2021-03-25	1.0	Sandip Saha Joy	Created initial version
2021-10-18	1.1	David Pasternak	Updated instructions

© IBM Corporation 2021. All rights reserved.