



# 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.

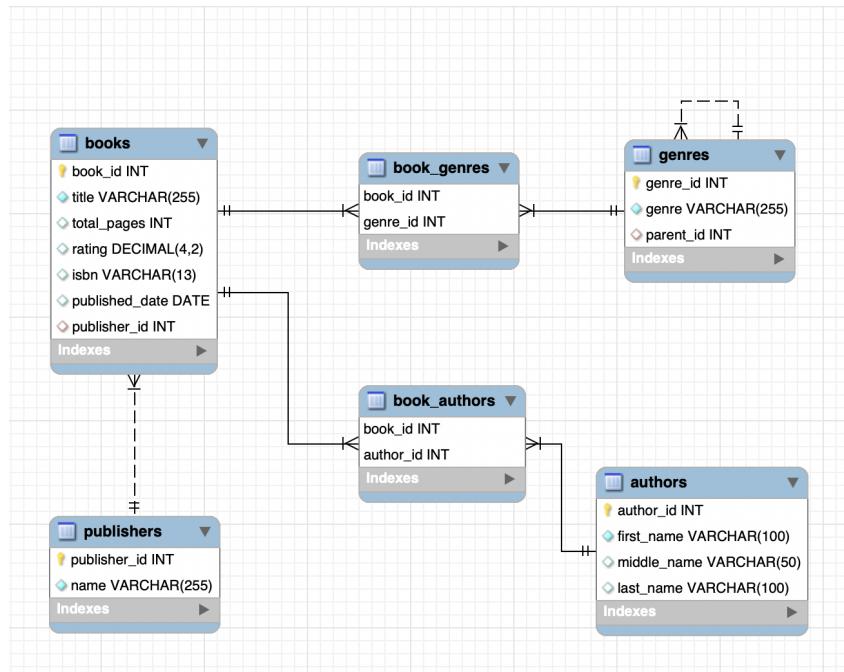


To complete this lab you will utilize the PostgreSQL relational database service available as part of IBM Skills Network Labs (SN Labs) Cloud IDE. SN Labs is a virtual lab environment used in this course.

## 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.

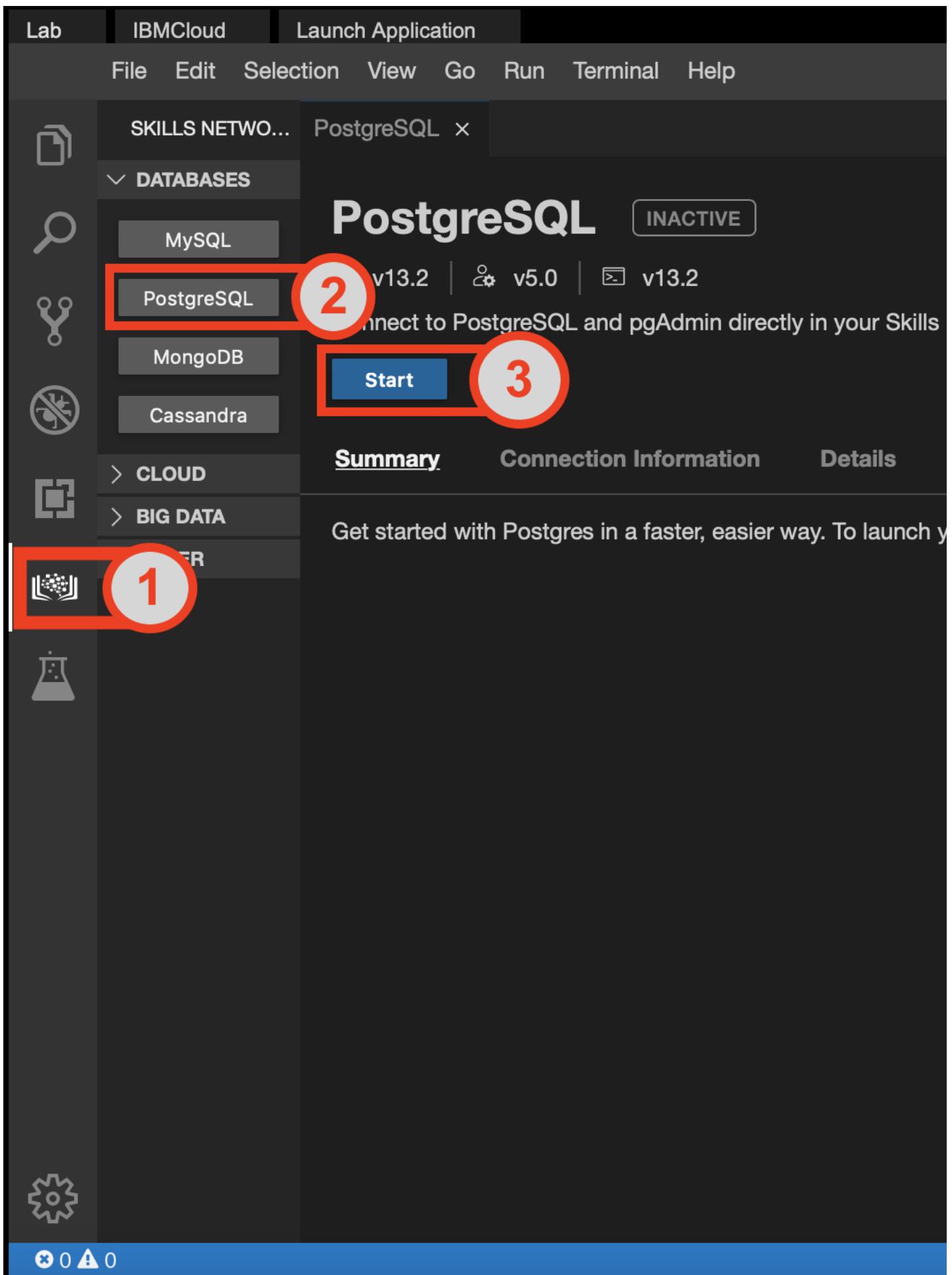
## Lab Structure

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.
  - [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.

PostgreSQL x

**Stop**

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

Your database and pgAdmin server are now ready to use and available with the IP address `127.0.0.1`. To learn more about the service or how to navigate PostgreSQL, please check out the Details section.

**Username:**  [Copy](#)

**Password:**  [Copy](#)

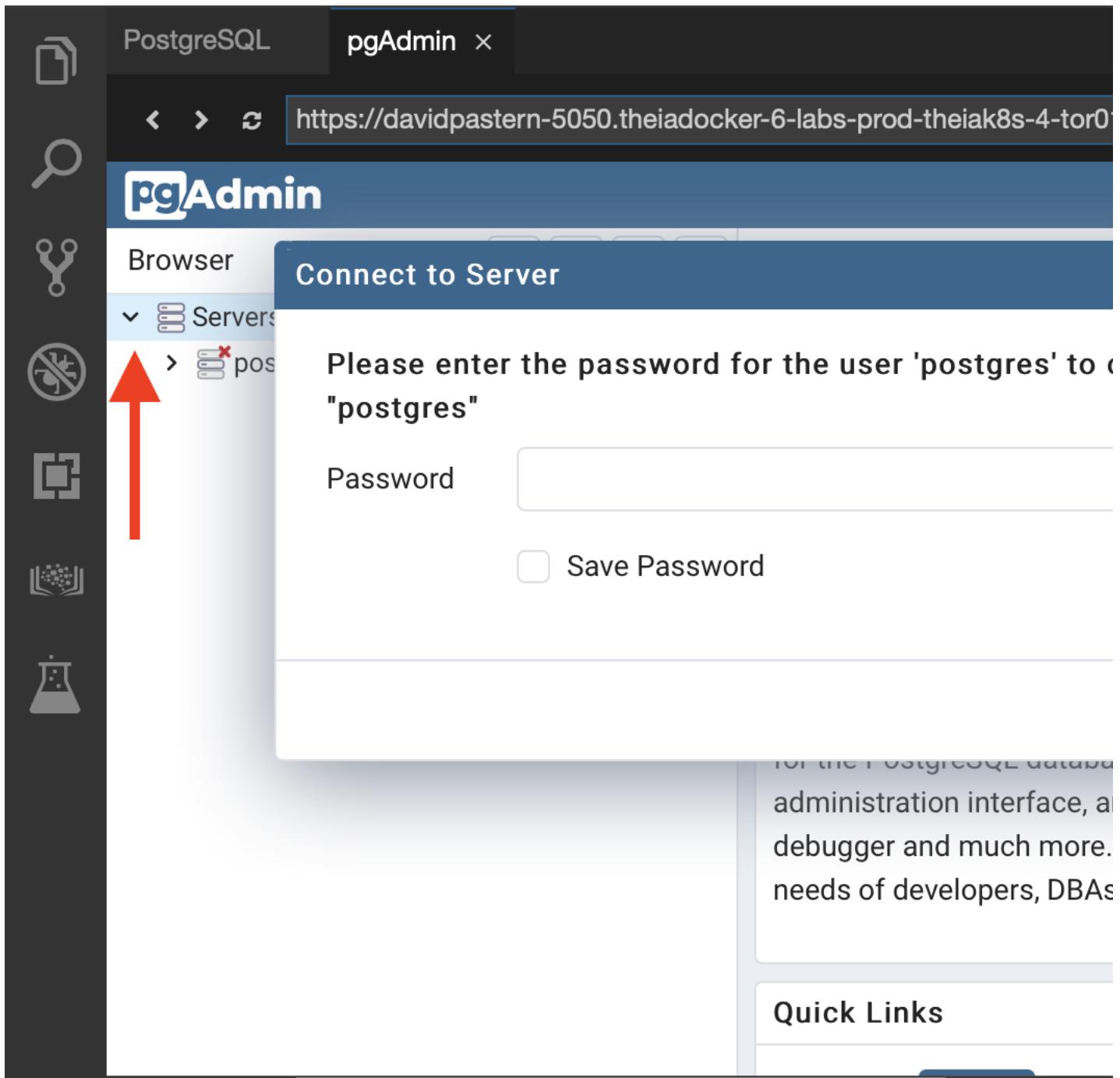
You can manage PostgreSQL via:

**pgAdmin** [Open](#)

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

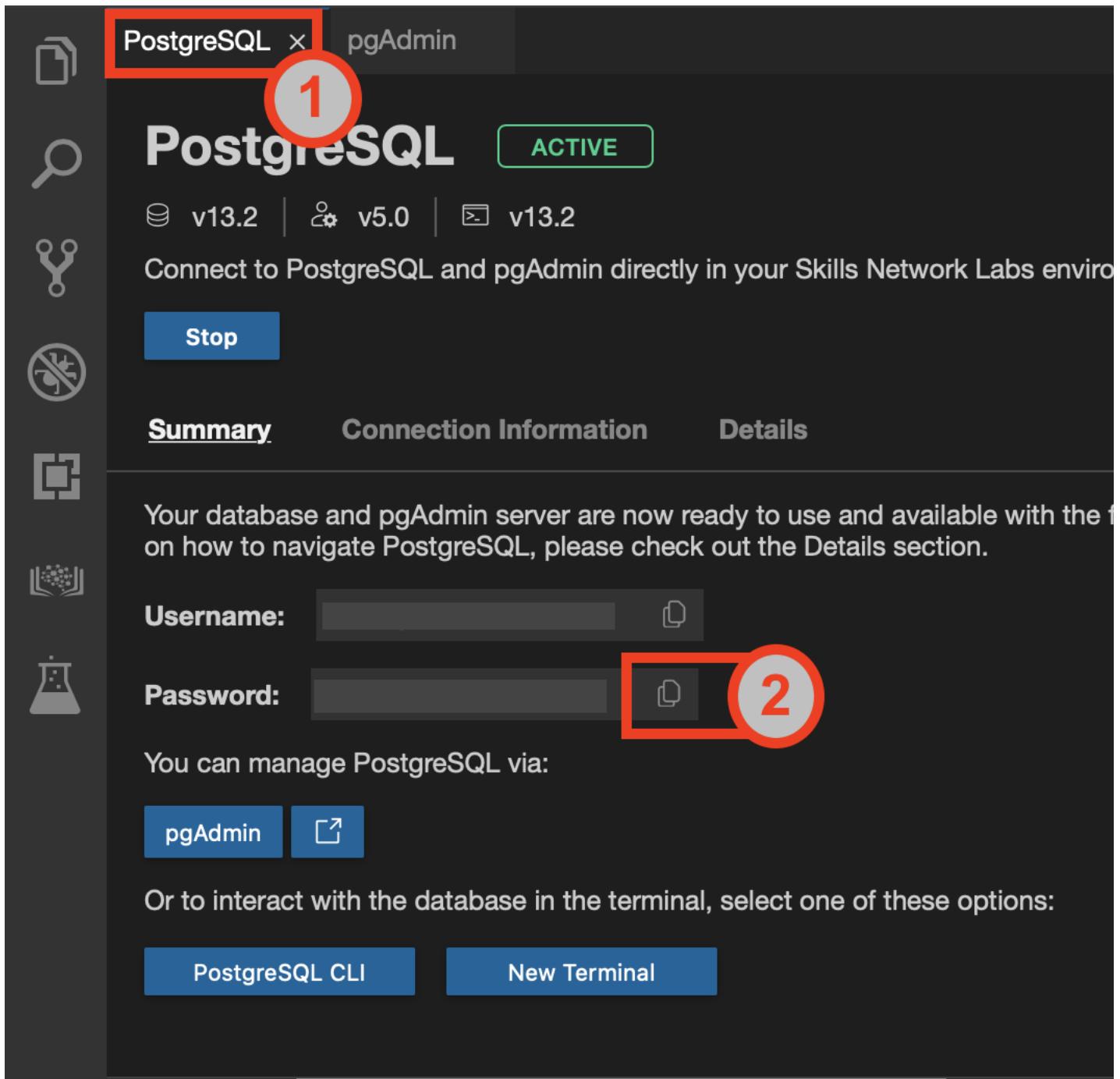
**PostgreSQL CLI**   **New Terminal**

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.



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.



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

# pgAdmin

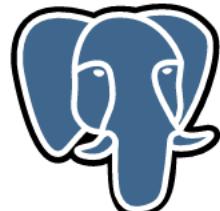
File ▾ Object ▾ Tools ▾ Help ▾

Browser

Servers

Dashboard Properties SQL

## Welcome



pgAdn  
Management

Feature rich | Maximise

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

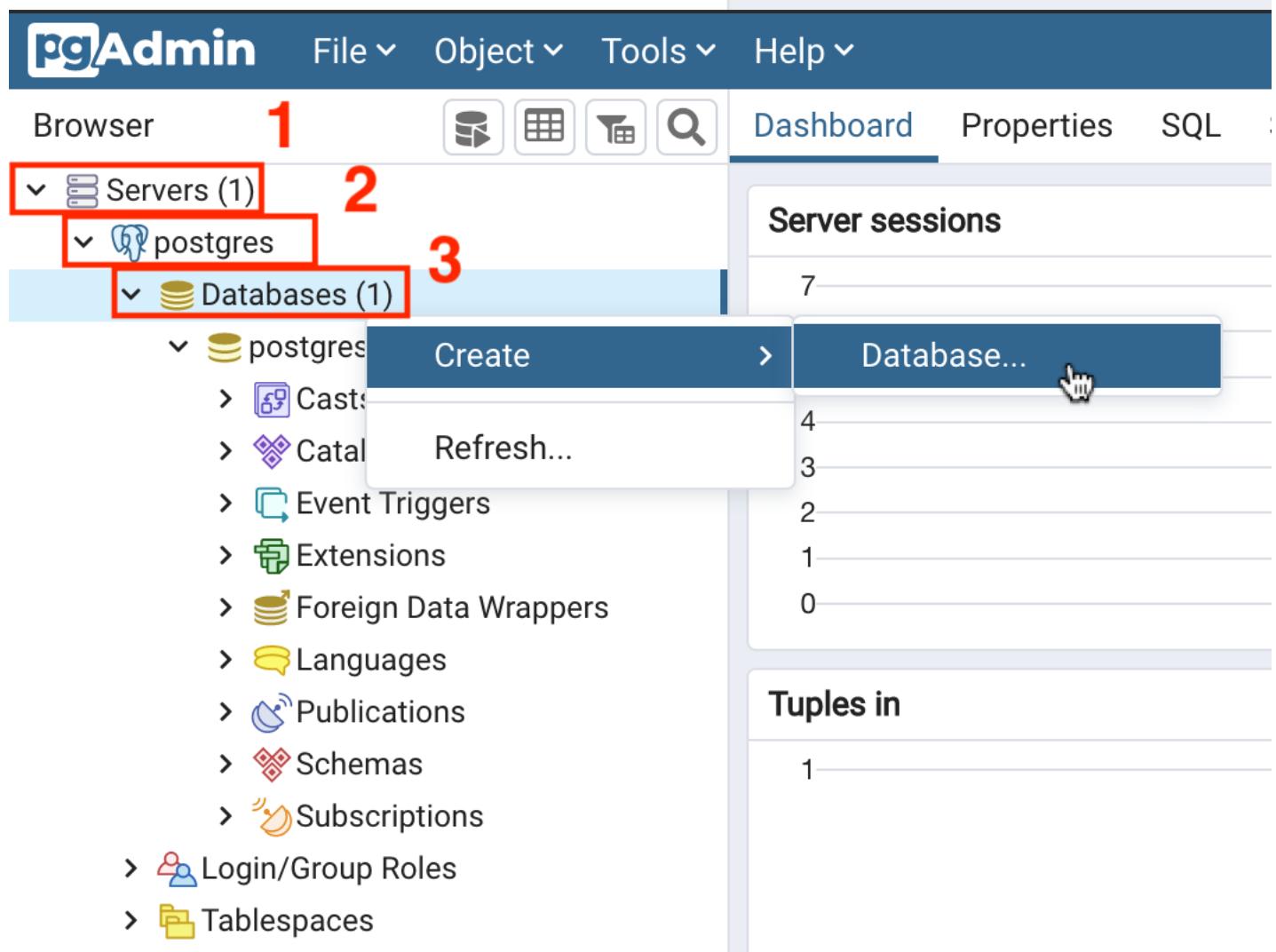
## Quick Links

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



## Create - Database

General Definition Security Parameters Advanced SQL

Database

eBooks

Owner

postgres

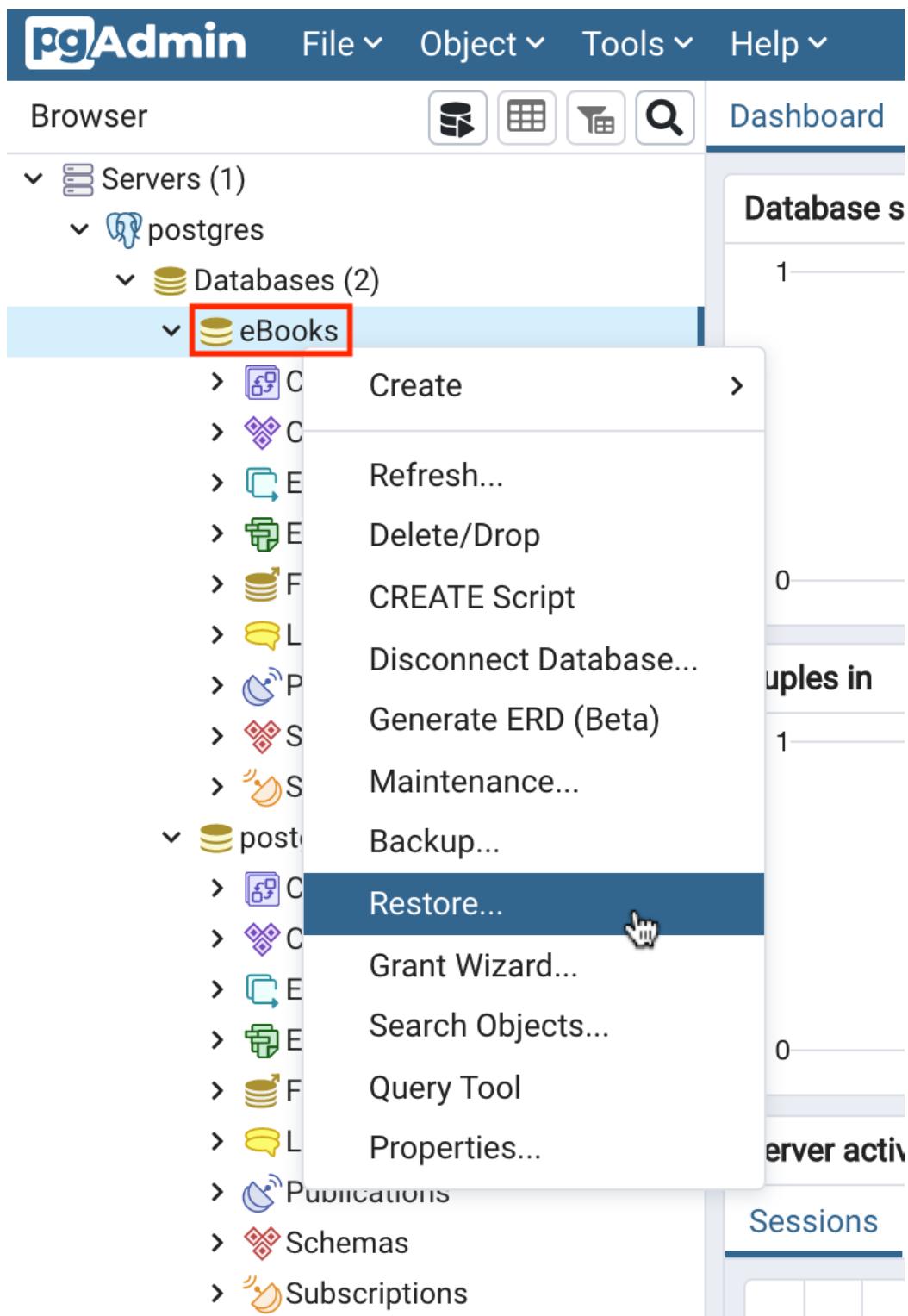
Comment

i

?

Cancel

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



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

Number of jobs

Role name

Select an item...



- Click the **Upload File** button.

## 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 from your local computer storage.

## 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 the X button.

## 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?

- 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			
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 **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** 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



### 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.

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

Create

View...

Refresh...



2. On the **General** tab, type **publisher\_and\_rating\_view** as name of the view. Then switch to **Code** tab.

Create - View

General Definition Code Security SQL

Name: publisher\_and\_rating\_view

Owner: postgres

Schema: public

Comment:

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

3. On the **Code** tab, copy and paste the code below. 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 - 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 on **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

Database sessions  
1

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

Server activity  
Sessions   Locks

PID
83

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

[Refresh...](#)

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			
	<b>title</b> character varying (255)	<b>rating</b> numeric (4,2)	<b>name</b> 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 on Materialized Views and go to Create > Materialized View.

pgAdmin   File ▾   Object ▾   Tools ▾   Help ▾

Browser   Dashboard   Properties

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

4 Materialized Views

- Procedures
- Sequences
- Tables (6)
- Trigger Functions
- Types
- Views (1)

Subscriptions

Create

Refresh...

Grant Wizard...

Search Objects...

Query Tool

Materialized

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 code below. 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 = pul  
3
```



× Car

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 a database navigation interface with a sidebar and a main content area.

**Left Sidebar:**

- 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

**Right Sidebar:**

- Gene
  - Name
  - OID
  - Owner
  - System
  - Comm
- Secul
  - Privileg
- Stora
  - Tables
  - Storage

**Content Area:**

1. Materialized Views (1)

2. publisher\_and\_rating\_materialized\_view

3. Refresh View

4. With data

Contextual menu for the Materialized View:

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

5. Right-click on **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

Create

Refresh...

Delete/Drop

Drop Cascade

Scripts

Refresh View

View/Edit Data

Search Objects...



## General

Name

OID

Owner

System

Common

## Security

Privileges

## Storage

Tables

Storage

All Rows

First

6. You will access the materialized view you created.

COLUMNS

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

	<b>title</b> character varying (255)	<b>rating</b> numeric (4,2)	<b>name</b> 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.