

Hands-on Lab: Views in PostgreSQL



Skills
Network

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.



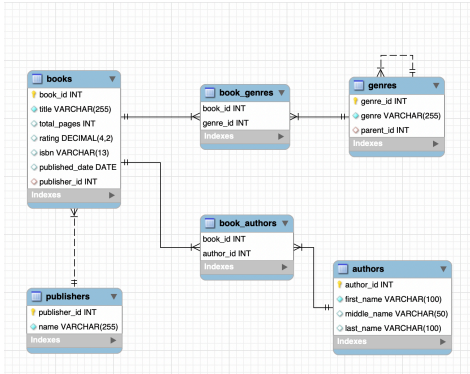
PostgreSQL

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

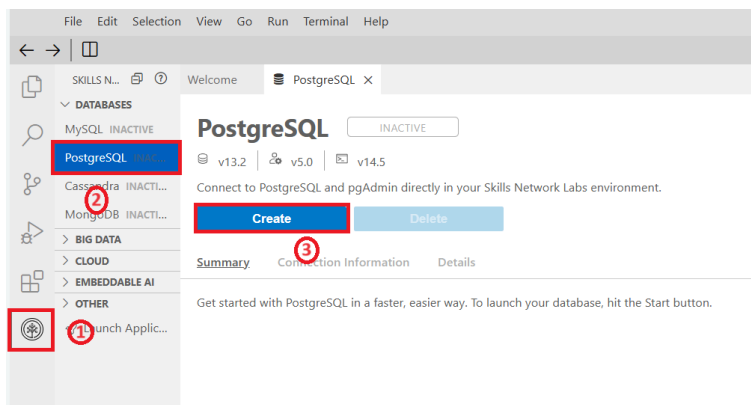
Lab structure

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.

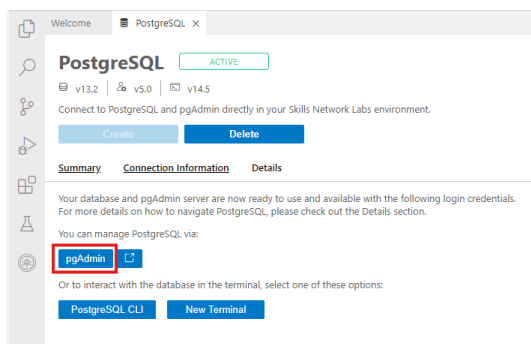
Task A: Restore a database schema and data

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:

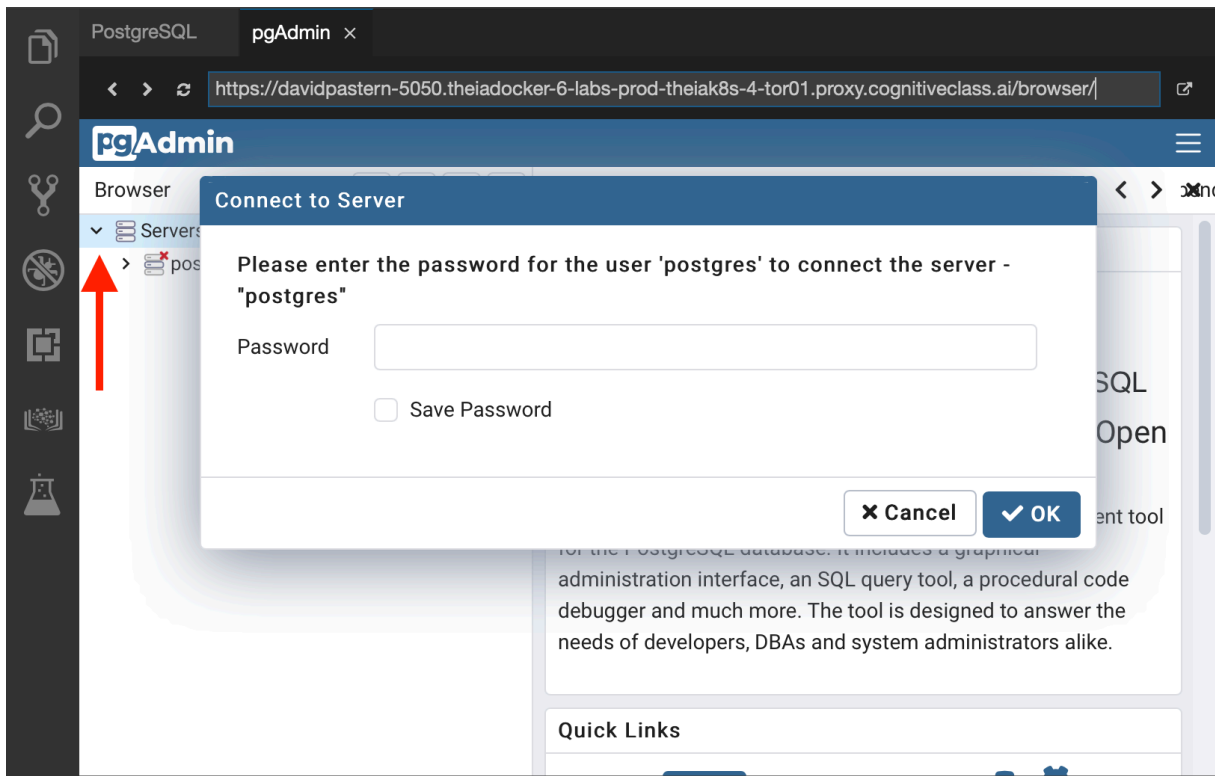
1. Download the following **eBooks** PostgreSQL dump file (containing the eBooks database schema and data) to your local computer.
 - [eBooks_pgsql_dump.tar](#)
2. Click the Skills Network extension button on the left side of the window.
3. Open the **DATABASES** menu and click **PostgreSQL**.
4. Click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



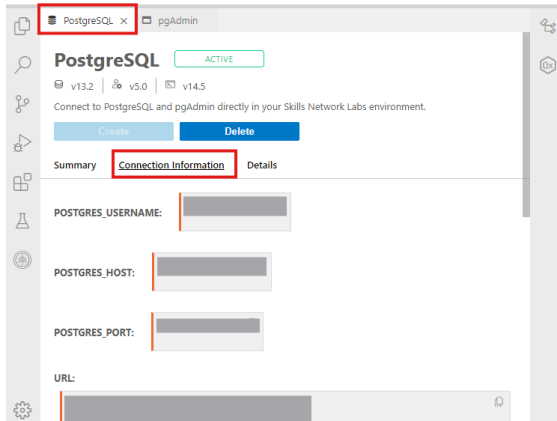
5. Next, open the pgAdmin Graphical User Interface by clicking **pgAdmin** in the Cloud IDE interface.



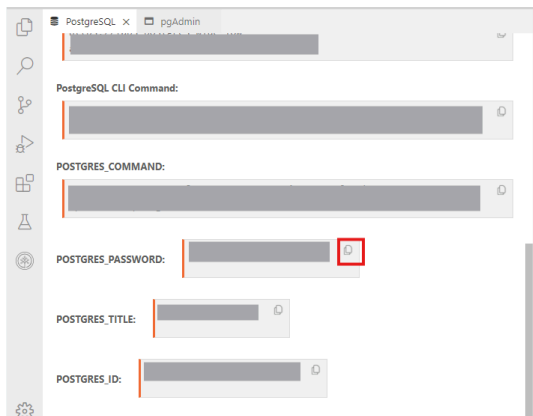
6. Once the pgAdmin GUI opens, click **Servers** tab on the left side of the page. You will be prompted to enter a password.



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



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



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.cognitiveclass.ai/browser/

pgAdmin File ▾ Object ▾ Tools ▾ Help ▾

Browser Servers

Dashboard Properties SQL Statistics Dependencies Dependents

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 is designed to answer the needs of developers, DBAs and system administrators alike.

Quick Links




Add New Server

Getting Started



PostgreSQL Documentation



pgAdmin Website

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**.

pgAdmin File ▾ Object ▾ Tools ▾ Help ▾

Browser Servers (1) postgres Databases (1)

1 2 3

postgres

- Cast
- Catal
- Event Triggers
- Extensions
- Foreign Data Wrappers
- Languages
- Publications
- Schemas
- Subscriptions
- Login/Group Roles
- Tablespaces

Create > Database...

Refresh...

Server sessions

7
4
3
2
1
0

Tuples in

1

Create - Database

×

General

Definition

Security

Parameters

Advanced

SQL

Database

eBooks

Owner

postgres

▼

Comment

?

?

×

Cancel

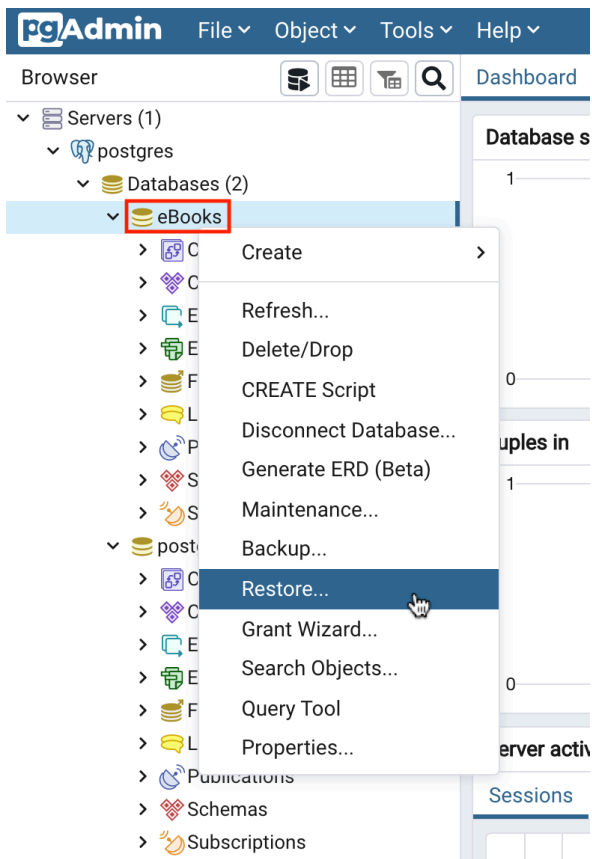
↺

Reset

💾

Save

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)

×

General

Data Options

Query Options

Table Options

Options

Format

Custom or tar

Filename

📁

Number of jobs

Role name

Select an item...

?

?

×

Close

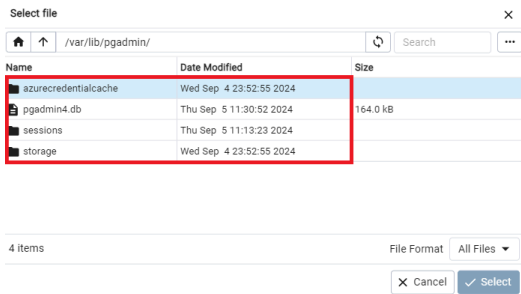
↺

Reset

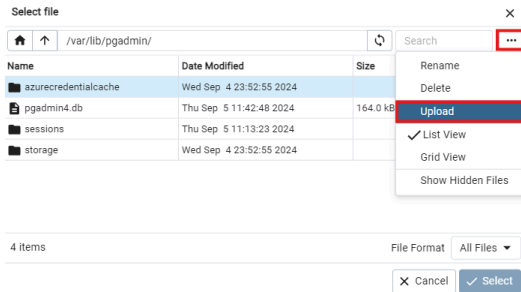
↗

Restore

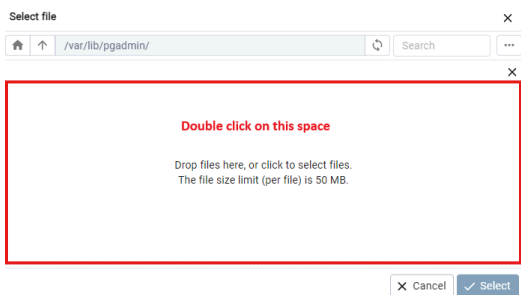
- Ensure that you upload the files to this path: `/var/lib/pgadmin/`. To do this, you can either manually navigate to the path (or) copy `/var/lib/pgadmin/`, replace `/home/` with it, and press Enter. You should then see some default files in that path, as shown below.



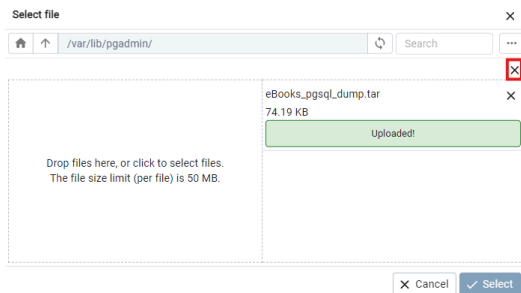
- Click on the three dots, then select **Upload**.



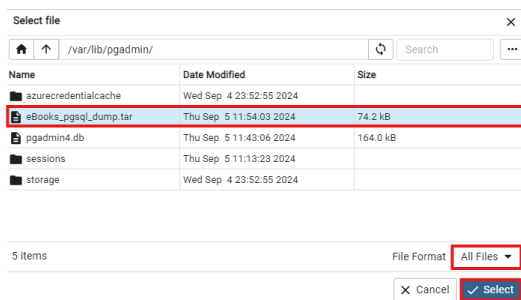
- Double-click on the drop files area and load the **eBooks_pgsql_dump.tar** you downloaded earlier on your local computer.



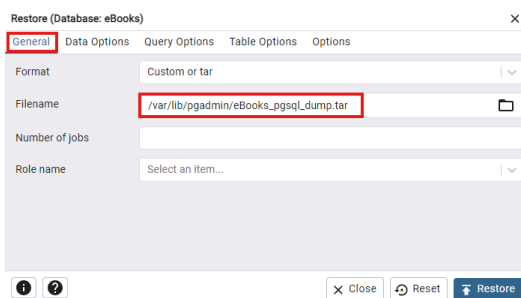
- When the upload is complete, close the drop files area by clicking **X**.



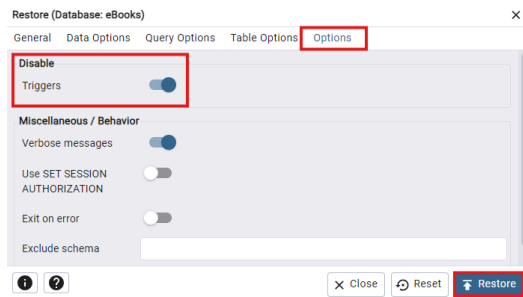
- Ensure **Format** is set to **All Files**, select the uploaded **eBooks_pgsql_dump.tar** file from the list, and then click **Select**.



- 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.

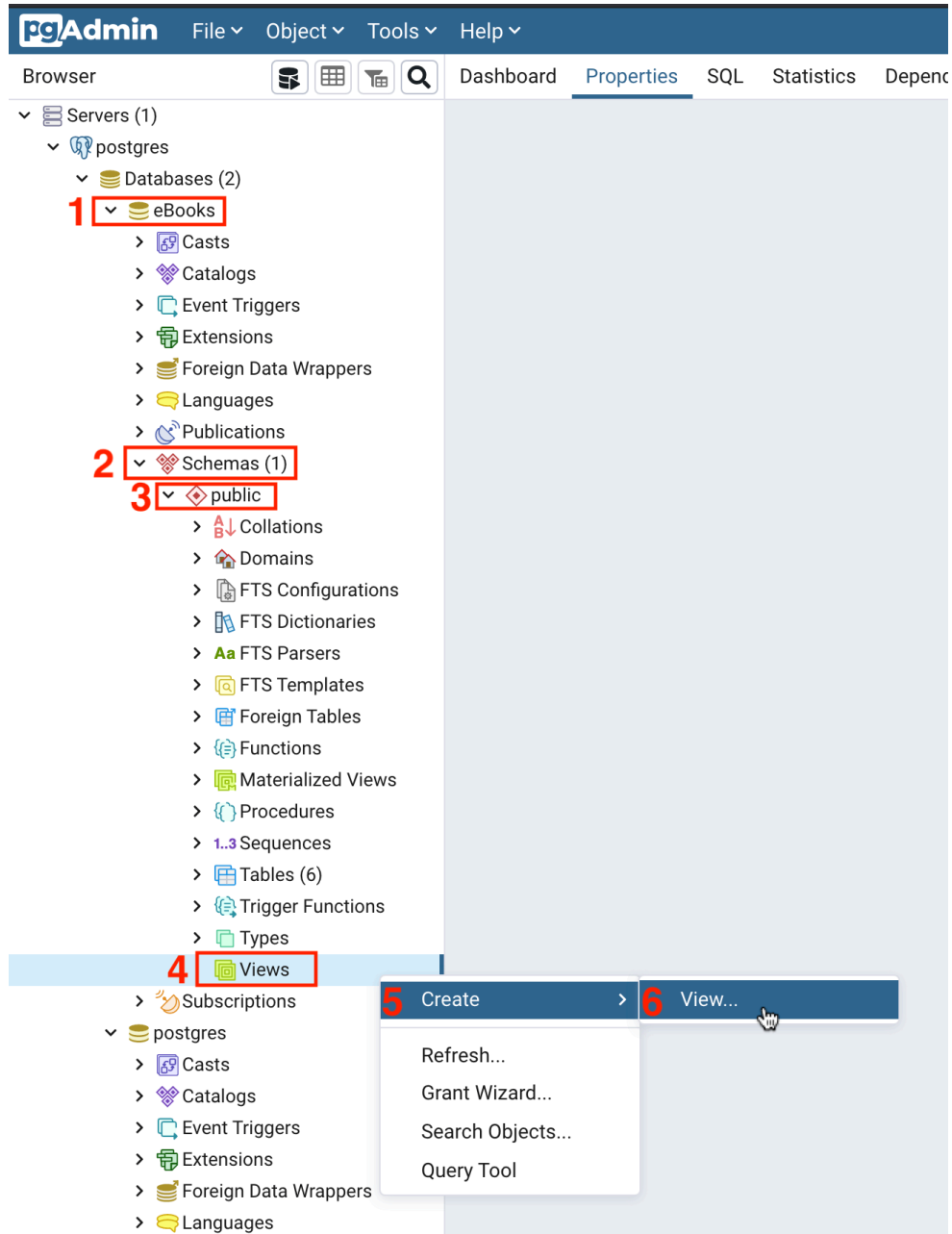


- Now switch to the **Options** tab. Under **Disable**, toggle on the **Triggers** option, and then click **Restore**.



Task B: Create and execute a view

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



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

i ?

Cancel

Reset

Save

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
```

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

i ?

Cancel

Reset

Save

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 SQL Statistics

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
 Views (1)
 publisher_and_rating view
 Columns
 Rules
 Triggers
 Subscriptions
 postgres
 Casts
 Catalogs
 Event Triggers
 Extensions
 Foreign Data Wrappers
 Languages
 Publications

Database sessions

1

0

Tuples in

Ins

18
16
14
12
10
8
6
4
2
0

Server activity

Sessions Locks Prepared Transactions

		PID	User	Age
✖	▶	83	postgres	p

1
2
3
4

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

All Rows
First 100 Rows
Last 100 Rows
Filtered Rows...

5. You will access the view you created. This action allows you to access and view the 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

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

File
Object
Tools
Help

Browser
Dashboard
Properties
SQL
Statistics

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

5

Create

6

Materialized View...
Refresh...
Grant Wizard...
Search Objects...
Query Tool

Sequences
Tables (6)
Trigger Functions
Types
Views (1)
Subscriptions

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

Create - Materialized View

General Definition Code Parameter Security SQL

Name publisher_and_rating_materialized_view

Owner postgres

Schema public

Comment

Close Reset Save

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
```

Create - Materialized View

General Definition Code Parameter Security SQL

1 SELECT books.title, books.rating, publishers.name

2 FROM books INNER JOIN publishers ON books.publisher_id = publishers.publisher_id

Close Reset Save

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

1 eBooks

- 2 Casts
- Catalogs
- Event Triggers
- Extensions
- Foreign Data Wrappers
- Languages
- Publications
- 3 Schemas (1)
 - public
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - 4 Materialized Views (1)
 - 5 publisher_and_rating_materialized_view
 - Columns
 - Indexes
 - Procedures
 - 1.3 Sequences
 - Tables (6)
 - Trigger Functions
 - Types
 - Views (1)
 - 6 publisher_and_rating_materialized_view
 - Columns
 - Rules
 - Triggers
 - Subscriptions

General

Name

OID

Owner

System materialized view?

Comment

Security

Privileges

Storage

Tablespace

Storage settings

Refresh View

With data

With no data

With data (concurrently)

With no data (concurrently)

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

pgAdmin

FileObjectToolsHelp

Browser

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

1.3 Sequences

Tables (6)

Trigger Functions

Types

Views (1)

publisher_and_rating

Columns

Rules

Triggers

Subscriptions

Create

Refresh...

Delete/Drop

Drop Cascade

Scripts

Refresh View

View/Edit Data

Search Objects...

Query Tool

Properties...

All Rows

First 100 Rows

Last 100 Rows

Filtered Rows...

Dashboard

Properties

SQL

i

General

Name

OID

Owner

System materialized view?

Comment

Security

Privileges

Storage

Tablespace

Storage settings

6. You will access the materialized view you created.

```
1 SELECT * FROM public.publisher_and_rating_materialized_view
2
```

	<div>title</div> <div>character varying (255)</div>	<div>rating</div> <div>numeric (4,2)</div>	<div>name</div> <div>character varying (255)</div>
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](#)

