



# Hands-on Lab : Create Tables and Load Data in PostgreSQL using pgAdmin

**Estimated time needed:** 20 minutes

In this lab, you will learn how to create tables and load data in the PostgreSQL database service using the pgAdmin graphical user interface (GUI) tool. The pgAdmin GUI provides an alternative to the command line for interacting with a PostgreSQL database using a graphical interface. This provides a number of key features for interacting with a PostgreSQL database in an easy to use format.

## Software Used in this Lab

In this lab, you will use [PostgreSQL Database](#). PostgreSQL is a Relational Database Management System (RDBMS) designed to efficiently store, manipulate, and retrieve 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

Books database has been used in this lab.

The following diagram shows the structure of the myauthors table from the Books database:

myauthors	
author_id	int
first_name	varchar(100)
middle_name	varchar(50)
last_name	varchar(100)

# Objectives

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

- Create databases and tables in a PostgreSQL instance
- Load data into tables manually using the pgAdmin GUI
- Load data into tables from a text/script file

## Lab Structure

In this lab, you will complete several tasks in which you will learn how to create tables and load data in the PostgreSQL database service using the pgAdmin graphical user interface (GUI) tool.

## Task A: Create a database

First, to create a database on a PostgreSQL server instance, you'll first want to actually launch a PostgreSQL server instance on Cloud IDE and open up the pgAdmin Graphical User Interface.

1. Click on the Skills Network extension button on the left side of the window.
2. Open the “DATABASES” drop down menu and click on “PostgreSQL”
3. Click on the “Start” button. PostgreSQL may take a few moments to start.



SKILLS NETWO...



## ▼ DATABASES



MySQL



PostgreSQL



MongoDB

## &gt; CLOUD

## &gt; BIG DATA



Cassandra

PostgreSQL x

# PostgreSQL

INACTIVE

2

v13.2 | v5.0 | v13.2

Start

3

Summary

Connection Information

Details

Get started with Postgres in a faster, easier way. To launch your database

1



✖ 0 ⚠ 0

4. Next, open the pgAdmin Graphical User Interface by clicking the “pgAdmin” button in the Cloud IDE interface.

[Stop](#)**Summary****Connection Information****Details**

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

**Username:****Password:**

You can manage PostgreSQL via:

[pgAdmin](#)

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

[PostgreSQL CLI](#)[New Terminal](#)

- 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.

PostgreSQL

pgAdmin x



https://davidpastern-5050.theiadocker-6-labs-prod-theiak8s-4-tor01.proxy.co

**pgAdmin**

Browser

Servers

pos

## Connect to Server

Please enter the password for the user 'postgres' to connect  
"postgres"

Password



Save Password



for the PostgreSQL database. It includes an administration interface, an SQL query tool, a debugger and much more. The tool is designed to meet the needs of developers, DBAs and sys

## Quick Links

6. To retrieve your password, click on the “PostgreSQL” tab near the top of the interface.
7. Click on the Copy icon to the left of your password to copy the session password onto your clipboard.



PostgreSQL x pgAdmin

1

# PostgreSQL

ACTIVE

DB v13.2 | User v5.0 | Databases 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 credentials. For more information on how to navigate PostgreSQL, please check out the Details section.

Username:

Password:

You can manage PostgreSQL via:

pgAdmin



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

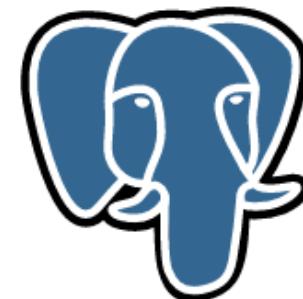
Or to interact with the database in the terminal, select one of these options.

PostgreSQL CLI

New Terminal

8. Navigate back to the “pgAdmin” tab and paste in your password, then click OK

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

**pgAdmin**File Object Tools HelpBrowserDashboardPropertiesSQLStatistics> Servers**Welcome****pgAdmin**

Management Tools

Feature rich | Maximises Pos

pgAdmin is an Open Source administration  
is designed to answer the needs of develop**Quick Links**

## Getting Started



PostgreSQL Documentation

10. In the tree-view, expand **Servers** > **postgres** > **Databases**. If prompted, enter your PostgreSQL service session password. Right-click on **Databases** and go to **Create** > **Database**. In the **Database** box, type **Books** as the name for your new database, and then click **Save**. Proceed to Task B.

Browser

1



Dashboard

Properties

SQL

Statistics

Servers (1)

2

postgres

3

Databases (1)

postgres

Create

Casts

Catalog

Refresh...

Event Triggers

Extensions

Foreign Data Wrappers

Languages

Publications

Schemas

Subscriptions

Login/Group Roles

Tablespaces

## Server sessions

7

Database...



4

3

2

1

0

## Tuples in

1



## Create - Database

General    Definition    Security    Parameters    Advanced    SQL

Database

Books

Owner

 postgres

Comment

 i ?

✖ Cancel

⟳ Rese

## Task B: Create tables

Now that you have your PostgreSQL service active and have created the **Books** database using pgAdmin, let's go ahead and create a few tables to populate the database and store the data that we wish to eventually upload into it.

1. In the tree-view, expand **Books > Schemas > public**. Right-click on **Tables** and go to **Create > Table**.

Browser



Dashboard

Properties

SQL

Statistics

Servers (1)

postgres

Databases (2)

Books 1

Casts

Catalogs

Event Triggers

Extensions

Foreign Data Wrappers

Languages

Publications

Schemas (1) 2

public 3

Collations

Domains

FTS Configurations

FTS Dictionaries

## Database sessions

1

0

## Tuples in

1

Ir

- > [Aa FTS Parsers](#)
- > [FTS Templates](#)
- > [Foreign Tables](#)
- > [Functions](#)
- > [Materialized Views](#)
- > [Procedures](#)
- > [1..3 Sequences](#)

4

- > [Tables](#)

- > [Trigger](#)

- > [Types](#)

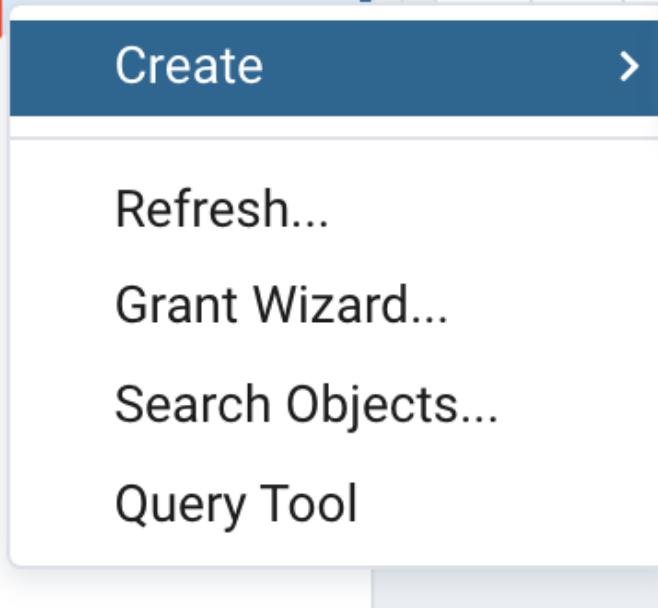
- > [Views](#)

- > [Subscriptions](#)

- > [postgres](#)

- > [Login/Group Roles](#)

- > [Tablespaces](#)



0

## Server activity

Sessions

Locks

Prepared Transaction

	PID	User

2. On the **General** tab, in the **Name** box, type **myauthors** as name of the table. Don't click Save, proceed to the next step.

## Create - Table

General

Columns

Advanced

Constraints

Partitions

Parameters

Security

SQ

Name

myauthors

Owner

postgres

Schema

public

Tablespace

Select an item...

Partitioned table?

No

Comment

 i ? × Cancel ↺ Rese

3. Switch to tab **Columns** and click the **Add new row** button four times to add 4 column placeholders. Don't click Save, proceed to the next step.

## Create - Table

General

Columns

Advanced

Constraints

Partitions

Parameters

Security

SQ

Inherited from table(s)

Select to inherit from...

### Columns

	Name ▾	Data type	Length/Precision	Scale	Not NULL?
		<input type="text"/>	<input type="button" value="Select an item..."/>		
		<input type="text"/>	<input type="button" value="Select an item..."/>		
		<input type="text"/>	<input type="button" value="Select an item..."/>		
		<input type="text"/>	<input type="button" value="Select an item..."/>		

 i ? × Cancel ↻ Rese

4. Enter the **myauthors** table definition structure information as shown in the image below in the highlighted boxes. Then click **Save**. Proceed to Task C.

## Create - Table

General Columns Advanced Constraints Partitions Parameters Security SQL

Inherited from table(s)

Select to inherit from...

### Columns

	Name	Data type	Length/Precision	Scale	Not NULL?
		author_id	integer		
		first_name	character varying	100	
		middle_name	character varying	50	
		last_name	character varying	100	



✗ Cancel

⟳ Rese

## Task C: Load data into tables manually using the pgAdmin GUI

Great! You now have a database and have created tables within it. With the pgAdmin GUI, you can insert values into the tables manually. This is useful if you have a few new entries you wish to add to the database. Let's see how to do it.

1. In the tree-view, expand **Tables**. Right-click on **myauthors** and go to **View/Edit Data > All Rows**.

Browser



Dashboard

Properties

SQL

Statistics

Servers (1)

postgres

Databases (2)

Books

Casts

Catalogs

Event Triggers

Extensions

Foreign Data Wrappers

Languages

Publications

Schemas (1)

public

Collations

Domains

FTS Config

FTS Dictionary

Type

Primary Key

Create &gt;

Refresh...

Count Rows

Delete/Drop

Drop Cascade

Reset Statistics

Import/Export...

Maintenance...

Scripts &gt;

1 **Tables (1)**

2 **myauthors**

> Truncate >

> Backup... >

> Restore... >

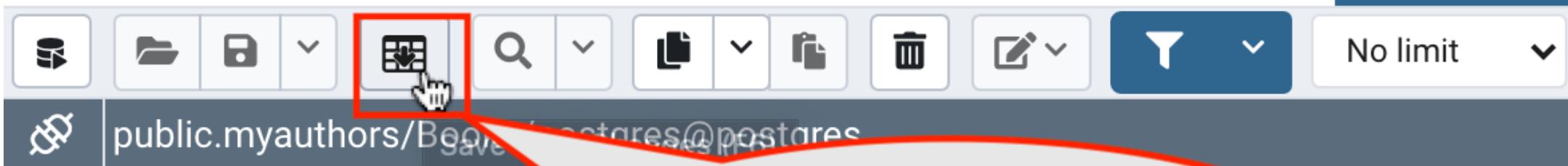
> View/Edit Data > **All Rows**

> Search Objects... > First 100 Rows

> Query Tool > Last 100 Rows

> Properties... > Filtered Rows...

2. You will insert 2 rows of data into the **myauthors** table. In the lower **Data Output** pane, enter **myauthors** table data information for 2 rows as shown in the highlighted boxes in the image below. Then click the **Save Data Changes** button. Proceed to Task D.

[Query Editor](#) [Query History](#)

```
1 SELECT * FROM public.myauthors  
2 ORDER BY author_id
```

**Save Data Changes  
icon**

Data Output   Explain   Messages   Notifications

	<b>author_id</b> [PK] integer	<b>first_name</b> character varying (100)	<b>middle_name</b> character varying (50)	<b>last_name</b> character varying
1		1 Merrit	[null]	Eric
2		2 Linda	[null]	Mul

## Task D: Load data into tables using a text/script file

In the previous task, you entered some data entries into a table manually with pgAdmin. While this method can be useful for small additions, if you wish to upload large amounts of data at once, that process becomes far too tedious. An alternative is to load data into tables from a text or script file containing the data you wish to enter. Let's take a look at how to do this.

- Finally, you will import the remainder of the **myauthors** table data from a csv text file. Download the csv file below to your local computer:
  - [myauthors.csv](#)
- In the tree-view, right-click on **myauthors** and go to **Import/Export**.

Browser



Dashboard Pr

▼ Servers (1)

▼ postgres

▼ Databases (2)

▼ Books

&gt; Casts

&gt; Catalogs

&gt; Event Triggers

&gt; Extensions

&gt; Foreign Data W

&gt; Languages

&gt; Publications

▼ Schemas (1)

▼ public

&gt; Collation

&gt; Domain

&gt; FTS Co

&gt; FTS Dict



Query Editor C

1    SELECT \*

R BY

Create &gt;

Refresh...

Count Rows

Delete/Drop

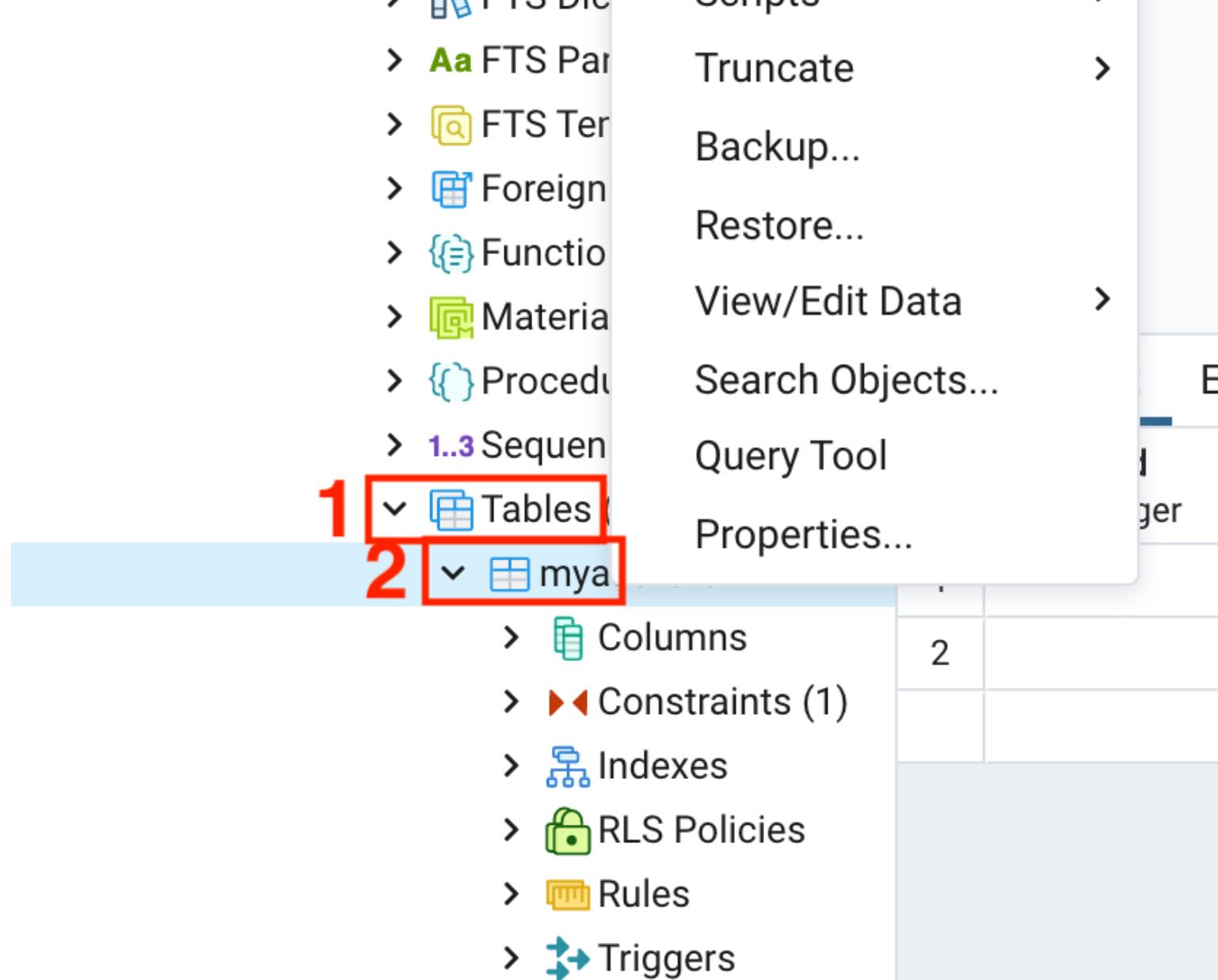
Drop Cascade

Reset Statistics

Import/Export... 

Maintenance...

Scripts &gt;



3. Follow the instructions below to import:

- Make sure Import/Export is set to **Import**, Format = **csv** and Header = **Yes**. Then click on the **Select file** button by the Filename box.

## Import/Export data - table 'myauthors'

Options Columns

Import/Export

Import

1

### File Info

Filename



Format

csv

2

Encoding

Select an item...

### Miscellaneous

OID

No

Header

Yes

3

Delimiter

Select from list...

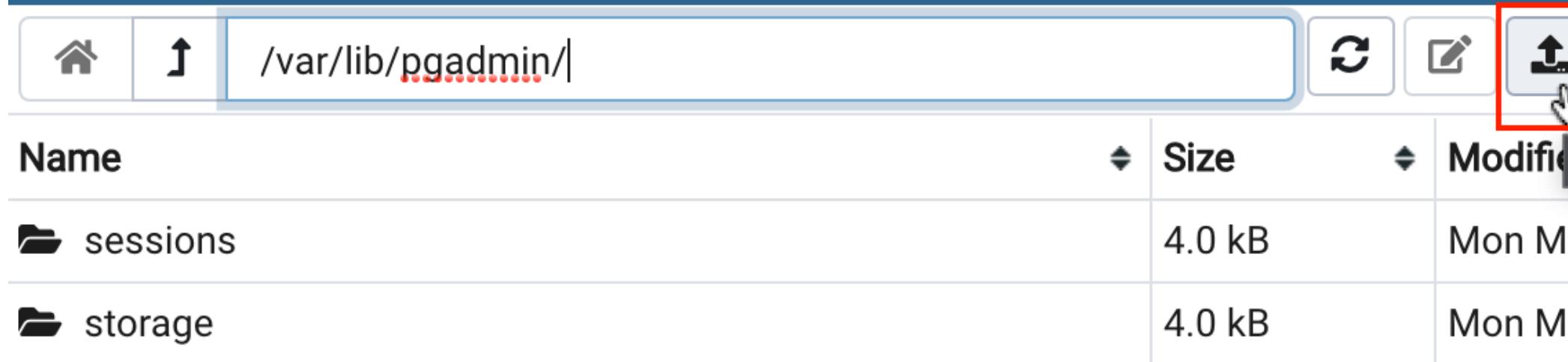
Specifies the character that separates columns within each row.

Specifies the character that separates columns within each row of the file. The default is a tab character in text format, a comma in CSV format, and a semicolon in DBF format. The separator character must be a single one-byte character. This option is not allowed for binary format.

X

- Click the **Upload File** button.

## Select file



The screenshot shows a file manager interface with a blue header bar containing the text "Select file". Below the header is a toolbar with several icons: a house, a refresh, an up arrow, a circular arrow, a pencil, and an upload icon. The upload icon is highlighted with a red box and has a small mouse cursor icon pointing to it. The main area displays a file list with the following columns: Name, Size, and Modified. The "Name" column is currently sorted by size (indicated by a downward arrow). There are two entries: "sessions" and "storage", both of which are 4.0 kB in size and were modified on Monday, March. The "Modified" column is also sorted by size.

Name	Size	Modified
sessions	4.0 kB	Mon M
storage	4.0 kB	Mon M

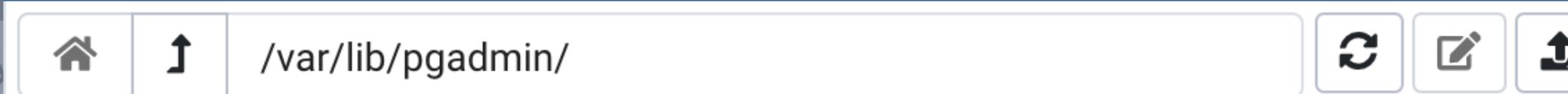
Show hidden files and folders?

F

**X** Ca

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

Select file



**Double click on this space**

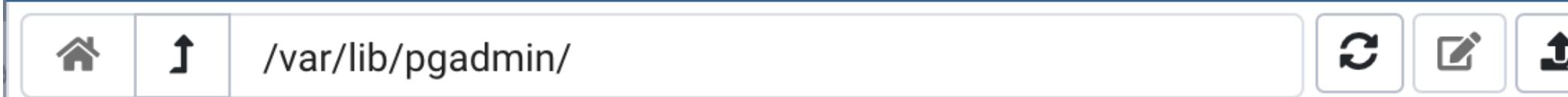
Drop files here to upload. The file size limit (per file) is 50 mb.

Show hidden files and folders?

**X** Ca

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

## Select file



26.6 KB



myauthors.csv

100%

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

Show hidden files and folders?

**X** Ca

- Select the uploaded **myauthors.csv** file from the list and click the **Select** button.

## Select file

		/var/lib/pgadmin/myauthors.csv			
Name			Size		Modified
myauthors.csv			26.0 kB		Mon Mar 1
sessions			4.0 kB		Mon Mar 1
storage			4.0 kB		Mon Mar 1

Show hidden files and folders?

F

**X** Ca

- Click **OK** and notification of import success should appear.

## Import/Export data - table 'myauthors'

Options    Columns

Import/Export

Import

### File Info

Filename

/var/lib/pgadmin/myauthors.csv

Format

csv

Encoding

Select an item...

### Miscellaneous

OID

No

Header

Yes

Delimiter

Select from list...

Specifies the character that separates columns within each row.

Specifies the character that separates columns within each row of the file. The default is a tab character in text format, a comma in CSV format. The character must be a single one-byte character. This option is not allowed for binary format.



## Import - Copying table data



Copying table data 'public.myauthors' on database 'Books' and server (postgres:5432)

Mon Mar 22 2021 02:26:40 GMT-0600 (Mountain Daylight Time)



0.02 seconds



More details...

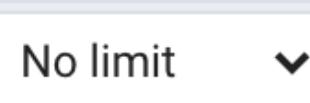


Stop Process



Successfully completed.

4. Repeat Task C Step 1 to check that the newly imported data rows appear along with your previously inserted 2 rows.



public.myauthors/Books/postgres@postgres

Query Editor Query History

```
1 SELECT * FROM public.myauthors
2 ORDER BY author_id ASC
```

Data Output Explain Messages Notifications

	author_id [PK] integer	first_name character varying (100)	middle_name character varying (50)	last_name character vary
1	1	Merrit	[null]	Eric
2	2	Linda	[null]	Mul
3	3	Alecos	[null]	Papadatos
4	4	Paul	C.van	Oorschot
5	5	David	[null]	Cronin
6	6	Richard	[null]	Blum
7	7	Yuval	Noah	Harari

		Fuval	Noah	Harlan
8	8	Paul	[null]	Albitz
9	9	David	[null]	Beazley
10	10	John	Paul	Shen
11	11	Andrew	[null]	Miller
12	12	Melanie	[null]	Swan
13	13	Neal	[null]	Ford
14	14	Nir	[null]	Shavit
15	15	Tim	[null]	Kindberg
16	16	Mike	[null]	McQuaid
17	17	Brian	P.	Hogan
18	18	Jean-Philippe	[null]	Aumasson
19	19	Lance	[null]	Fortnow
20	20	Richard	C.	Jeffrey
21	21	William	L.	Simon
22	22	Magnus	Lie	Hetland
23	23	Mike	[null]	McShaffry
24	24	Norman	[null]	Matloff

25	25	John	E.	Hopcroft
26	26	S.	[null]	Sudarshan

As you can see, the data contained in the **csv** file was successfully uploaded into the table and you did not have to manually input hundreds of entries.

## Conclusion

**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
2021-03-15	1.0	Sandip Saha Joy	Created initial version
2021-10-18	1.1	David Pasternak	Updated lab instructions

**© IBM Corporation 2020. All rights reserved.**