



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

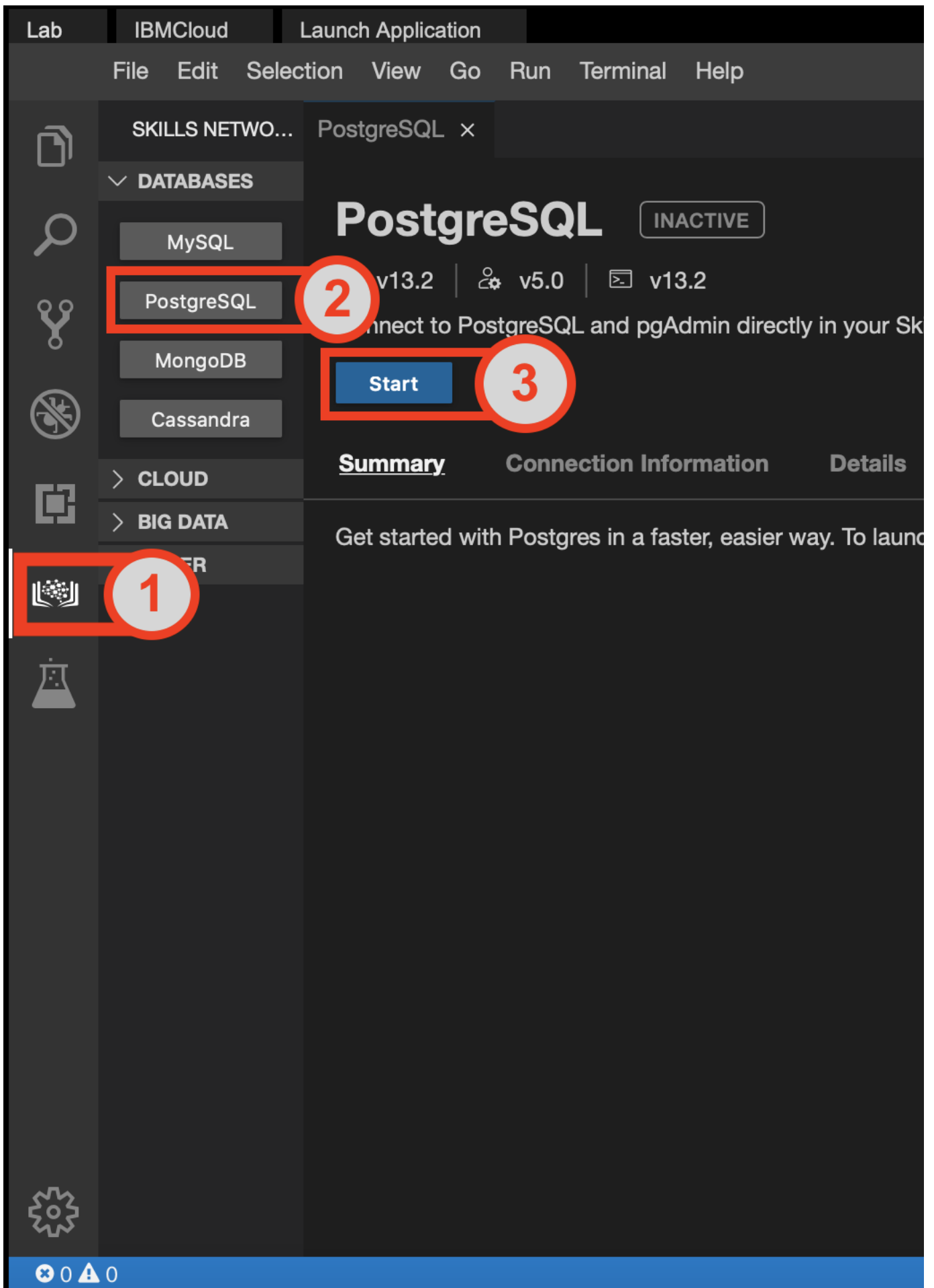
Lab Structure

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.

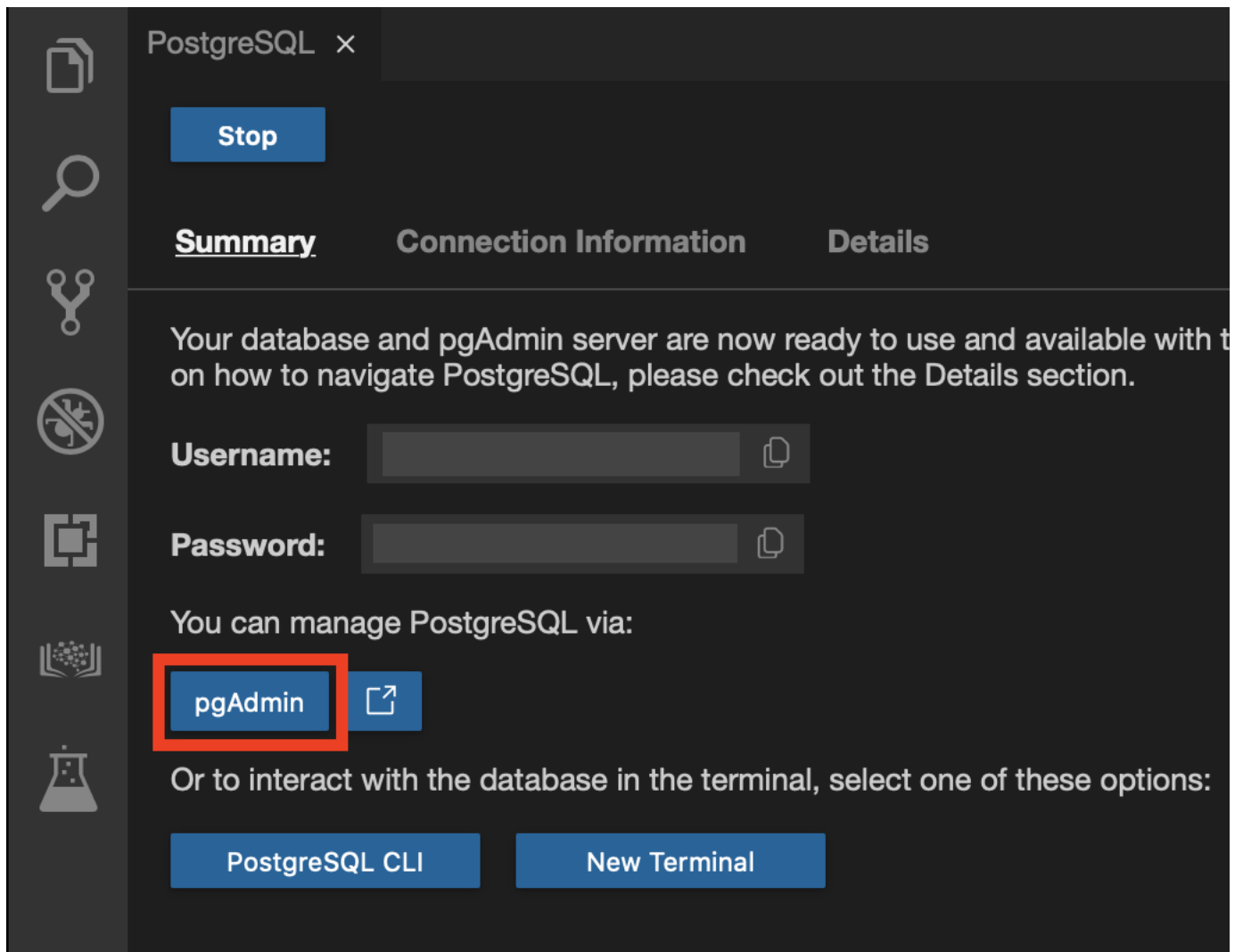
Task A: Create a database

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.

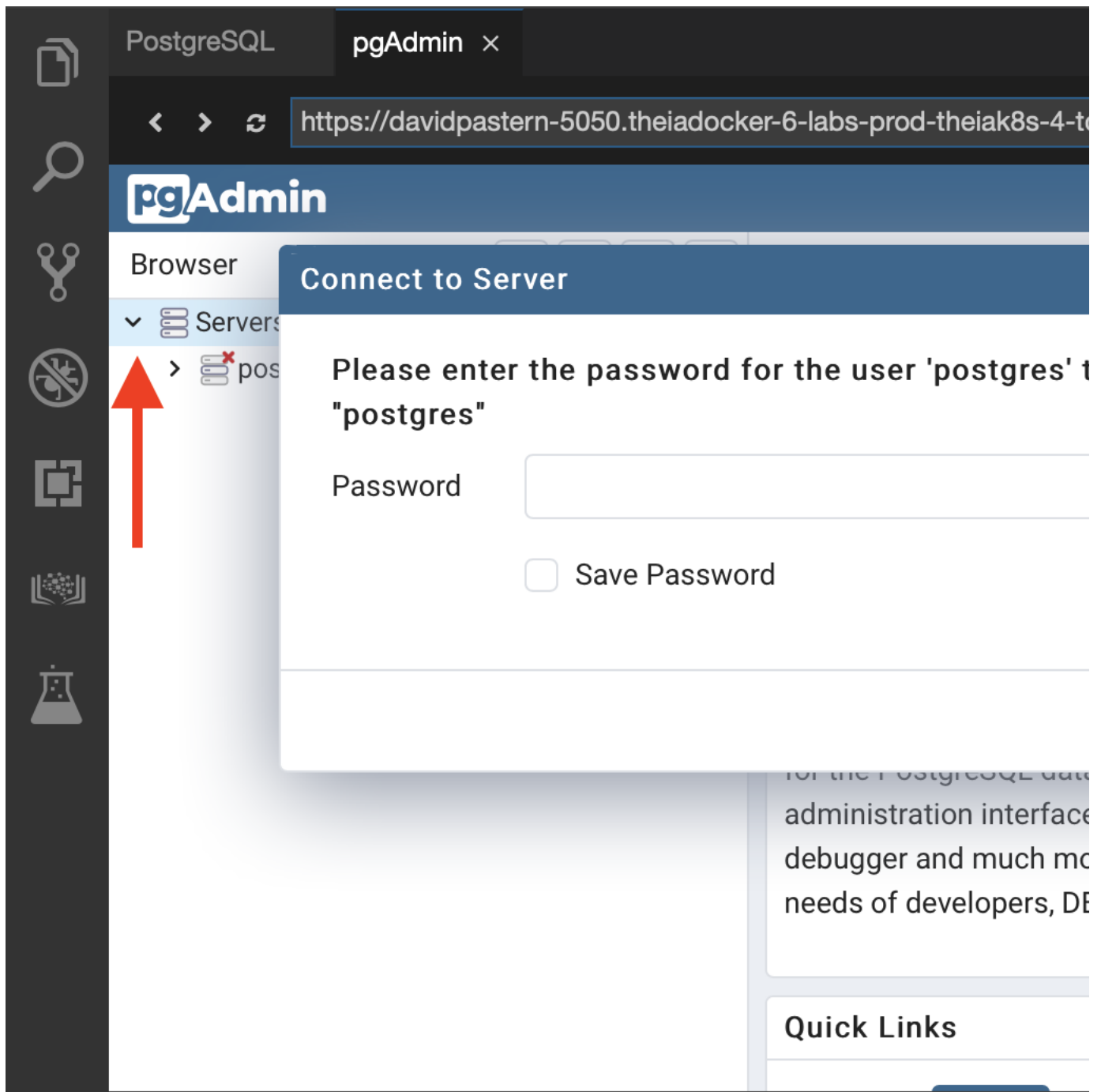
1. Click on the Skills Network extension button on the left side of the window.
2. Open the "DATABASES" drop down menu and click on "PostgreSQL"
3. Click on the "Start" button. PostgreSQL may take a few moments to start.



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



5. 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.



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

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 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:

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.

sandipsahajo-5050.theiadocker-27.proxy.cognitive

pgAdmin

File

Object

Tools

Help

Browser

Dashboard


Properties

SQL

>

Servers

Welcome



pgAd


Manageme

Feature rich | Maxim

pgAdmin is an Open Source ad
is designed to answer the need

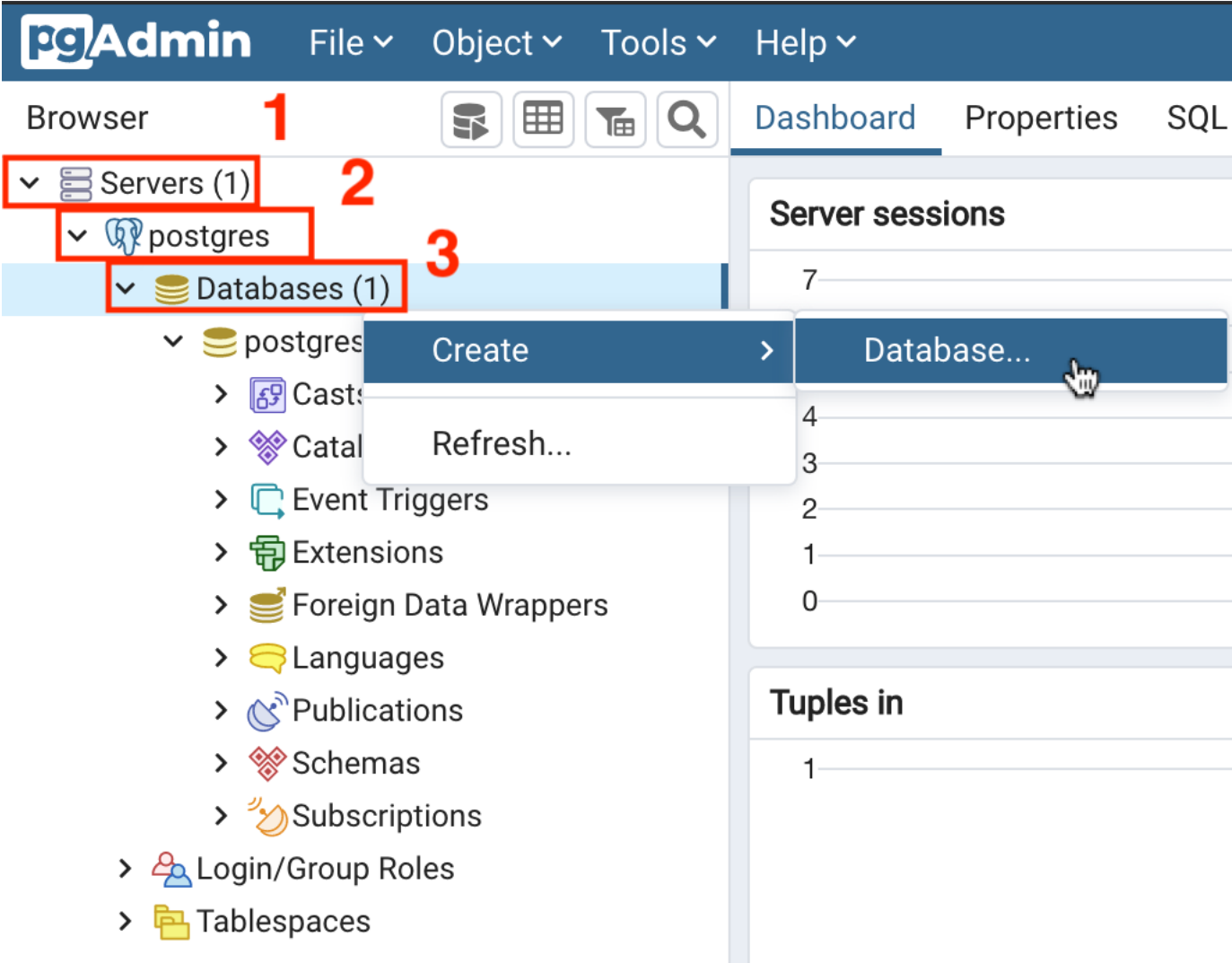
Quick Links

Getting Started



PostgreSQL Docur

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.



Create - Database

General

Definition

Security

Parameters


Advanced

SQL

Database

Books

Owner

 postgres

Comment

i

?

✕

Cancel

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**.

pgAdmin

File ▾Object ▾Tools ▾Help ▾

Browser

Servers (1)

postgres

Databases (2)

Books1

Casts

Catalogs

Event Triggers

Extensions

Foreign Data Wrappers

Languages

Publications

Schemas (1)2

public3

Collations

Domains

FTS Configurations

FTS Dictionaries

FTS Parsers

FTS Templates

Foreign Tables

Functions

Materialized Views

Procedures

1.3 Sequences

4

Tables

Trigger

Types

Views

Subscriptions

postgres

Login/Group Roles

Tablespaces

Dashboard

Properties

SQL

Database sessions

1

0

Tuples in

1

0

Server activity

SessionsLocksPrepared

PID

User

Create >

Table...

Refresh...

Grant Wizard...

Search Objects...

Query Tool

about:blank

9/25

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

Se

Name

myauthors

Owner

postgres

Schema

public

Tablespace

Select an item...

Partitioned table?

No

Comment

i

?

Cancel

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

Se

Inherited from table(s)

Select to inherit from...

Columns


		Name ▲	Data type	Length/Precision	Scale
		<input type="text"/>	<div>Select an item... ▼</div>		
		<input type="text"/>	<div>Select an item... ▼</div>		
		<input type="text"/>	<div>Select an item... ▼</div>		
		<input type="text"/>	<div>Select an item... ▼</div>		

i

?

✕ Cancel

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


Se


Inherited from table(s)

Columns

		Name	Data type	Length/Precision	Scale
		author_id	integer ▼		
		first_name	character varying ▼	100	
		middle_name	character varying ▼	50	
		last_name	character varying ▼	100	





 Cancel

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**.

pgAdmin

File ▾Object ▾Tools ▾Help ▾

Browser

Servers (1)

▼

postgres

▼

Databases (2)

▼

Books

▼

Casts

▸

Catalogs

▸

Event Triggers

▸

Extensions

▸

Foreign Data Wrap

▸

Languages

▸

Publications

▸

Schemas (1)

▼

public

▼

Collations

▸

Domains

▸

FTS Config

▸

FTS Diction

▸

FTS Parser

▸

FTS Templ

▸

Foreign Tal

▸

Functions

▸

Materialize

▸

Procedures

▸

Sequences

▸

Tables (1)

▼

Columns

▸

Constraints (1)

▸

Indexes

▸

RLS Policies

▸

Rules

▸

Triggers

▸

Dashboard

Properties

SQL

Type

Primary Key

Create ▸

Refresh...

Count Rows

Delete/Drop

Drop Cascade

Reset Statistics

Import/Export...

Maintenance...

Scripts ▸

Truncate ▸

Backup...

Restore...

View/Edit Data ▸

Search Objects...

Query Tool

Properties...

All Rows

First 100 Row

Last 100 Row

Filtered Rows

1

2

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.

The screenshot shows the DBeaver application window. At the top, there are tabs for 'Dashboard', 'Properties', 'SQL', 'Statistics', 'Dependencies', and 'Dependents'. Below these is a toolbar with various icons. One icon, representing a table with a downward arrow, is highlighted with a red rectangular box. A red callout bubble points to this icon with the text 'Save Data Changes icon'. Below the toolbar is a dark bar with the text 'public.myauthors/Bee...@postares'. Underneath that is a tabbed interface with 'Query Editor' and 'Query History'. The 'Query Editor' tab is active, showing a SQL query: '1 SELECT * FROM pub' and '2 ORDER BY author_id'. The bottom part of the image is a light gray area representing the query results.

	author_id [PK] integer	first_name character varying (100)	middle_name character varying (50)	last_name character varying (100)
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.

1. 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](#)
2. In the tree-view, right-click on **myauthors** and go to **Import/Export**.

The screenshot shows the pgAdmin 4 web interface. On the left, the 'Browser' pane displays a tree structure: Servers (1) > postgres > Databases (2) > Books > Schemas (1) > public > Tables. A red box highlights the 'Tables' entry, with a red '1' next to it. Another red box highlights the 'mya' database entry under 'Tables', with a red '2' next to it. A context menu is open over the 'Tables' entry, showing various actions. The 'Import/Export...' option is highlighted in blue, with a mouse cursor pointing at it. Other options in the menu include 'Create', 'Refresh...', 'Count Rows', 'Delete/Drop', 'Drop Cascade', 'Reset Statistics', 'Maintenance...', 'Scripts', 'Truncate', 'Backup...', 'Restore...', 'View/Edit Data', 'Search Objects...', 'Query Tool', and 'Properties...'. The right pane shows the 'Query Editor' with a SQL query starting with 'SELECT'.

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 with a file. The default is a tab character in text format, a comma in CSV format, and a pipe character in binary format.


- 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 **myauthors.csv** 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


Show hidden files and folders?☐


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

Select file



/var/lib/pgadmin/



26.6 KB

myauthors.csv



100%

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


Show hidden files and folders?☐




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

Select file



/var/lib/pgadmin/myauthors.csv



Name	Size
 myauthors.csv	26.0 kB
 sessions	4.0 kB
 storage	4.0 kB

Show hidden files and folders?☐

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

Import/Export data - table 'myauthors'

OptionsColumns

Import/ExportImport

File Info

Filename/var/lib/pgadmin/myauthors.csv

Formatcsv

EncodingSelect an item...

Miscellaneous

OIDNo

HeaderYes

DelimiterSelect from list...

Specifies the character that separates columns with file. The default is a tab character in text format, a must be a single one-byte character. This option is 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.

Dashboard

Properties

SQL

Statistics

Dependencies

Dependents

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

	<div>author_id</div> <div>[PK] integer</div>		<div>first_name</div> <div>character varying (100)</div>		<div>middle_name</div> <div>character varying (50)</div>	
1		1	Merrit		[null]	
2		2	Linda		[null]	
3		3	Alecos		[null]	
4		4	Paul		C.van	
5		5	David		[null]	
6		6	Richard		[null]	
7		7	Yuval		Noah	
8		8	Paul		[null]	
9		9	David		[null]	
10		10	John		Paul	
11		11	Andrew		[null]	
12		12	Melanie		[null]	
13		13	Neal		[null]	
14		14	Nir		[null]	
15		15	Tim		[null]	
16		16	Mike		[null]	
17		17	Brian		P.	
18		18	Jean-Philippe		[null]	
19		19	Lance		[null]	
20		20	Richard		C.	
21		21	William		I	

9/14/23, 1:32 PM

about:blank

21	21	William	E.
22	22	Magnus	Lie
23	23	Mike	[null]
24	24	Norman	[null]
25	25	John	E.
26	26	S.	[null]

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.