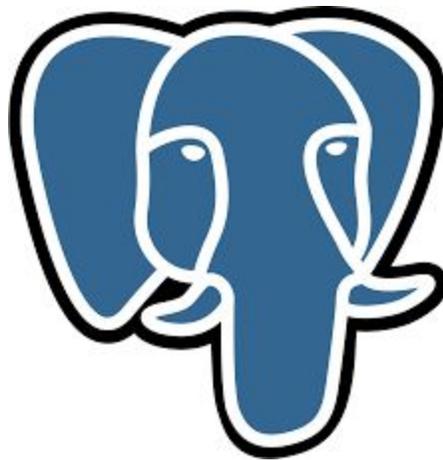




**How to create a shared database:
An example using PostGreSQL and
pgAdmin4**



A Shared Database allows you to store your documents in your favorite relational (SQL) database rather than on the file system. This means that multiple users can concurrently use the same synchronized storage location. Folders in a Shared Database will show up under the **Shared Databases** icon in the Sources panel, once the user has logged into the database.

A Shared Database can be used for everything a local database is used for. Note that unread status, agents and shared folders belong to individual users rather than the database. For example Bob may see a document as unread, but Joe will see that same document as read if he has read it.

For more information about Supported Database Systems and setup, please refer to section 19 of the [Geneious manual](#).

This document provides guidelines for setting up a shared database using PostgreSQL server using pgAdmin4, a GUI management system for the SQL database.

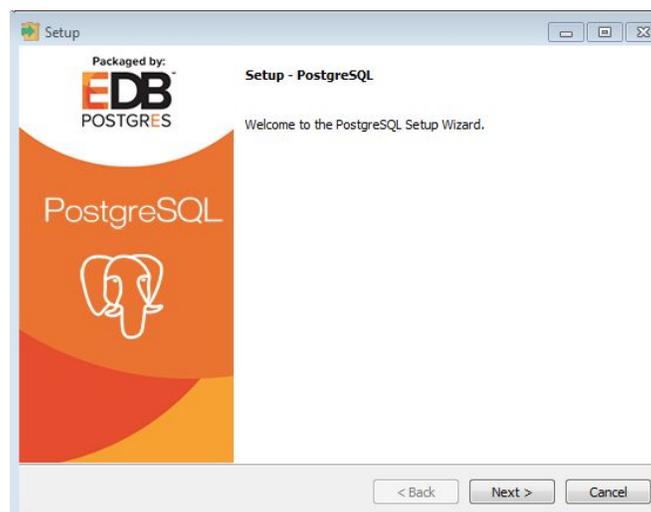
1. Installation of PostgreSQL and PgAdmin

Download and install PostgreSQL from the following link:

<http://www.enterprisedb.com/products-services-training/pgdownload>

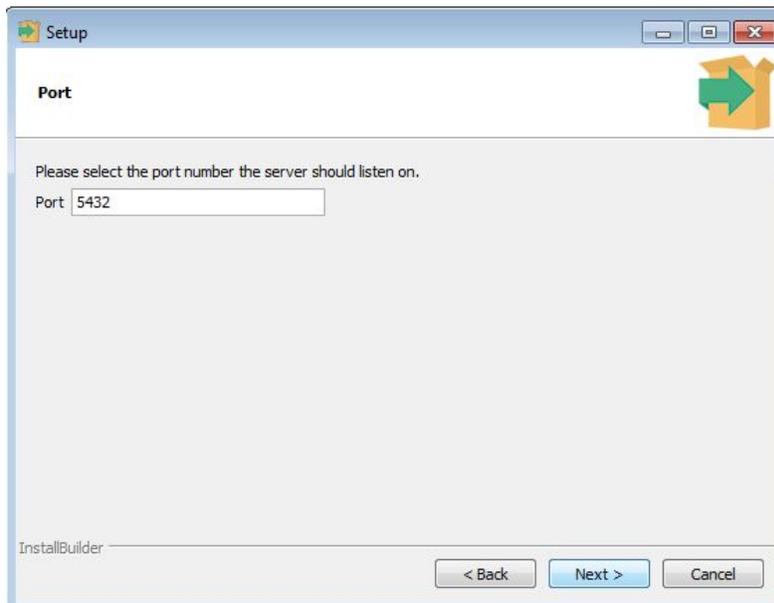
Run the executable once downloaded. This will install the **PostgreSQL server**, and **pgAdmin4**, the GUI database management system. Follow the setup instructions.

First the PostgreSQL server will be installed.

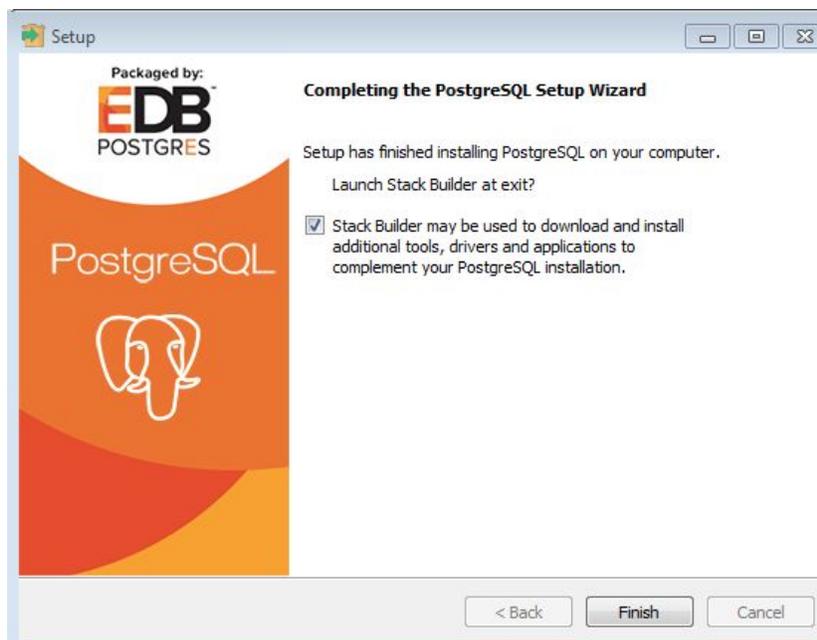


Follow the steps and provide a password for the database superuser when prompted.

Select the port number, leave the default or change it to another one if required by your network configuration:



Once the installation of PostgreSQL has finished, you can continue with the Stack Builder for adding tools, but in this example we will uncheck the option and finish the installation.



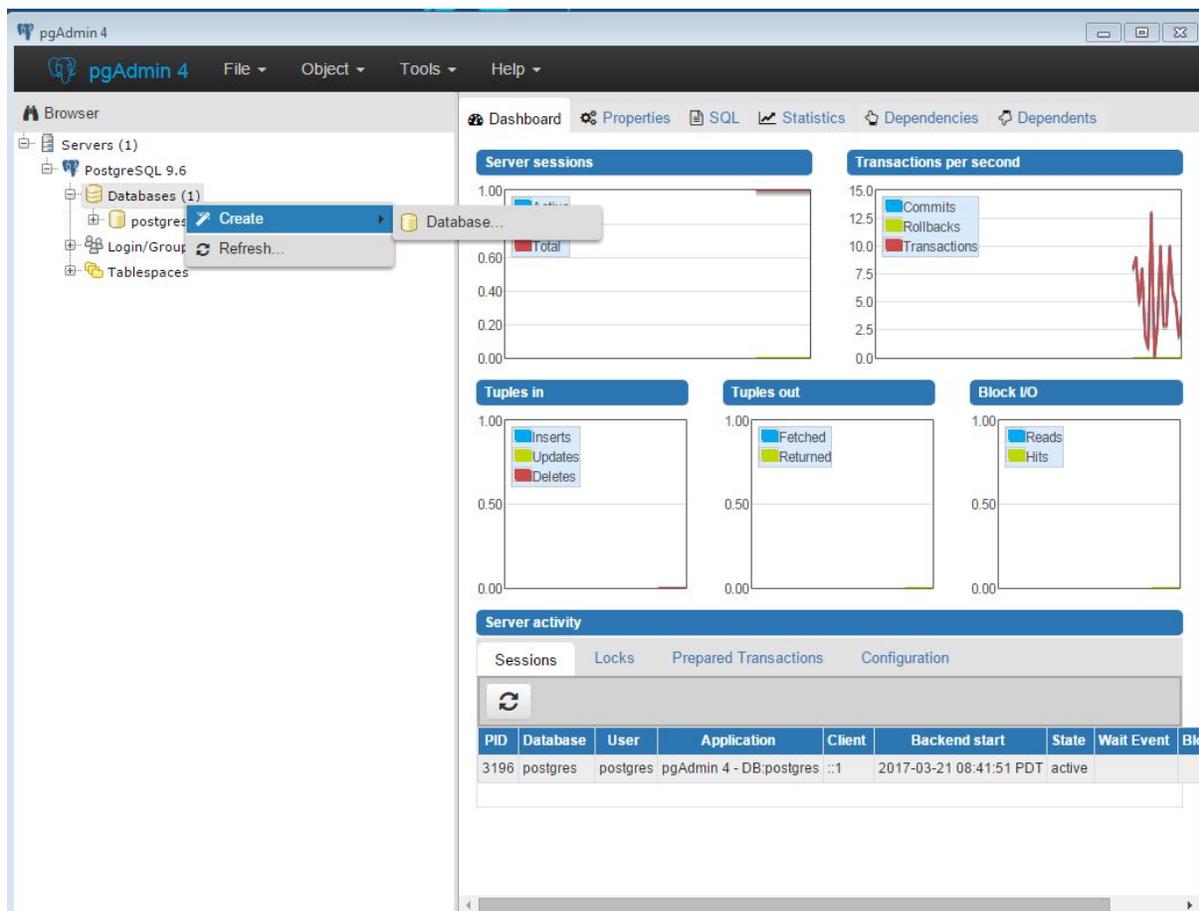
2. Creating a new database

The database management program, pgAdmin4, which was installed as part of the PostgreSQL installation, provides an interface for creating and managing databases and users.

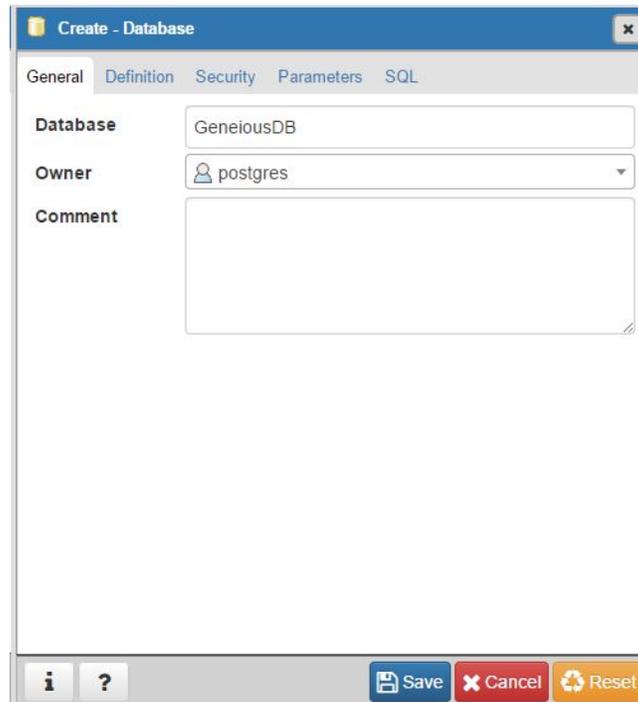
Note that pgAdmin4 v 3.0 and later will run within your default Web browser application

To create a new database, open pgAdmin4 and connect to the database by double clicking on PostgreSQL then providing the password for the superuser selected during the installation. If PostgreSQL is not immediately visible, Click **Browsers** → **Servers**.

Then create a new database by right clicking on Databases and Create → Database, as below:

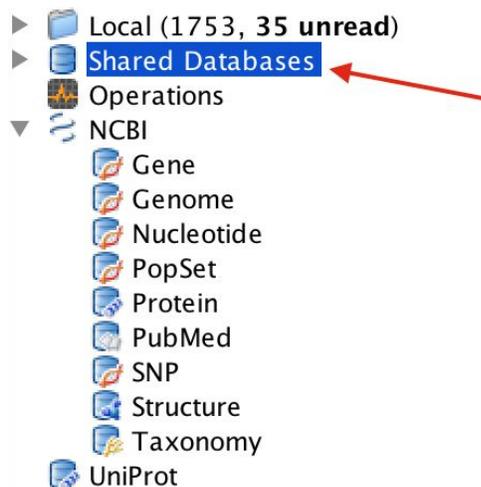


Provide a name for the new database, for example GeneiousDB, and make sure the superuser account (postgres in this case) is the owner of the database:

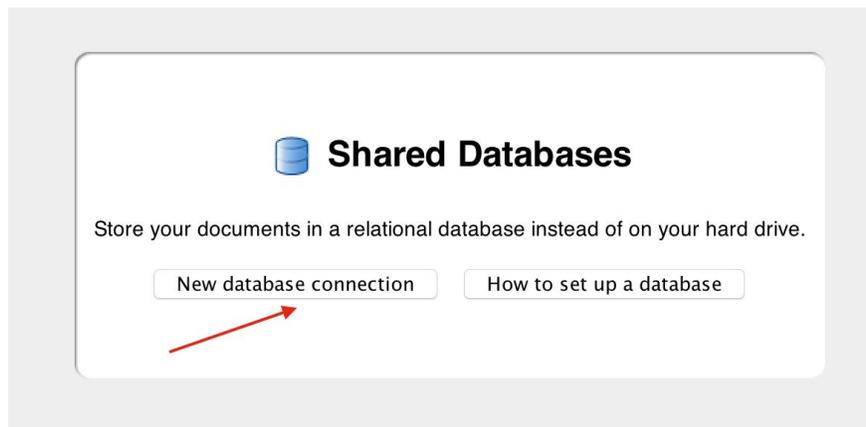


To finish setting up the database, open Geneious and connect to the newly created database

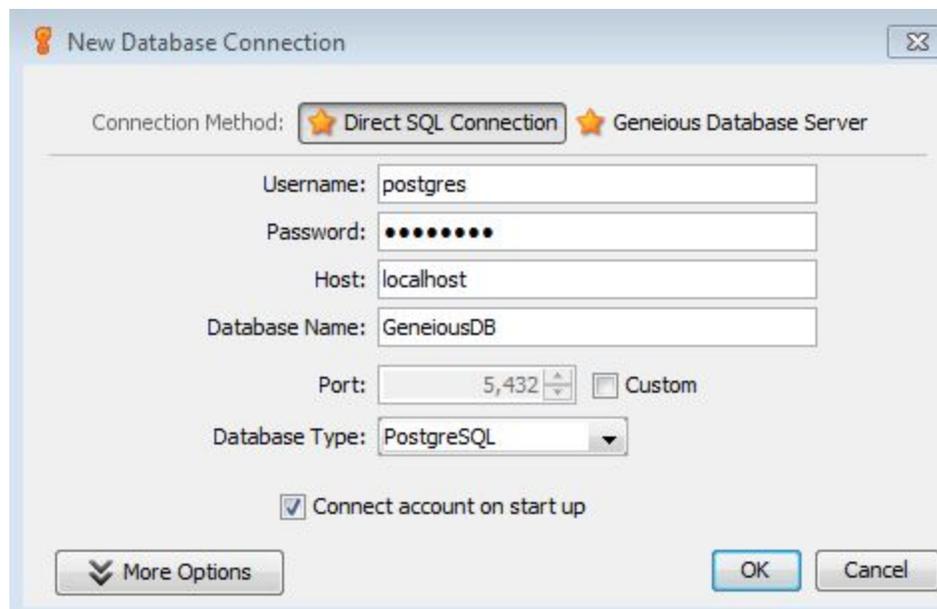
To do this, click on the Shared Database icon in the Geneious Sources panel:



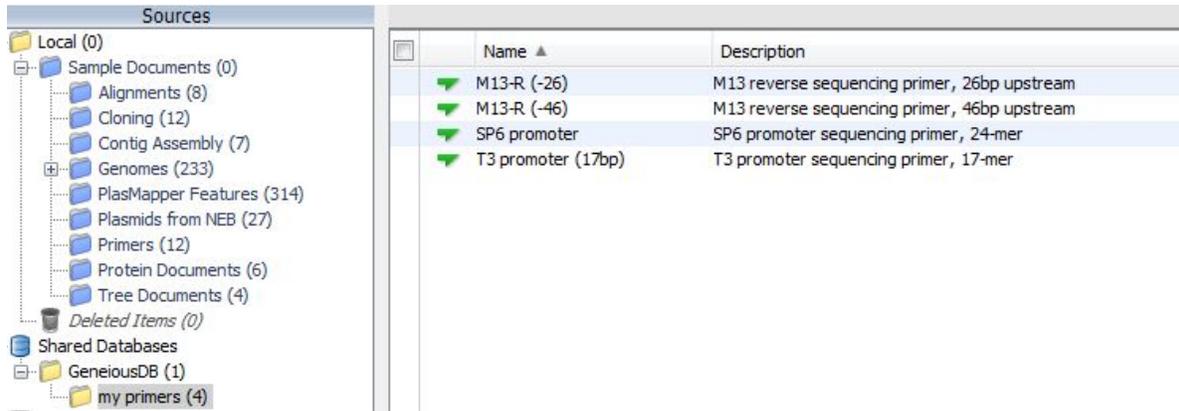
And select "New Database Connection"



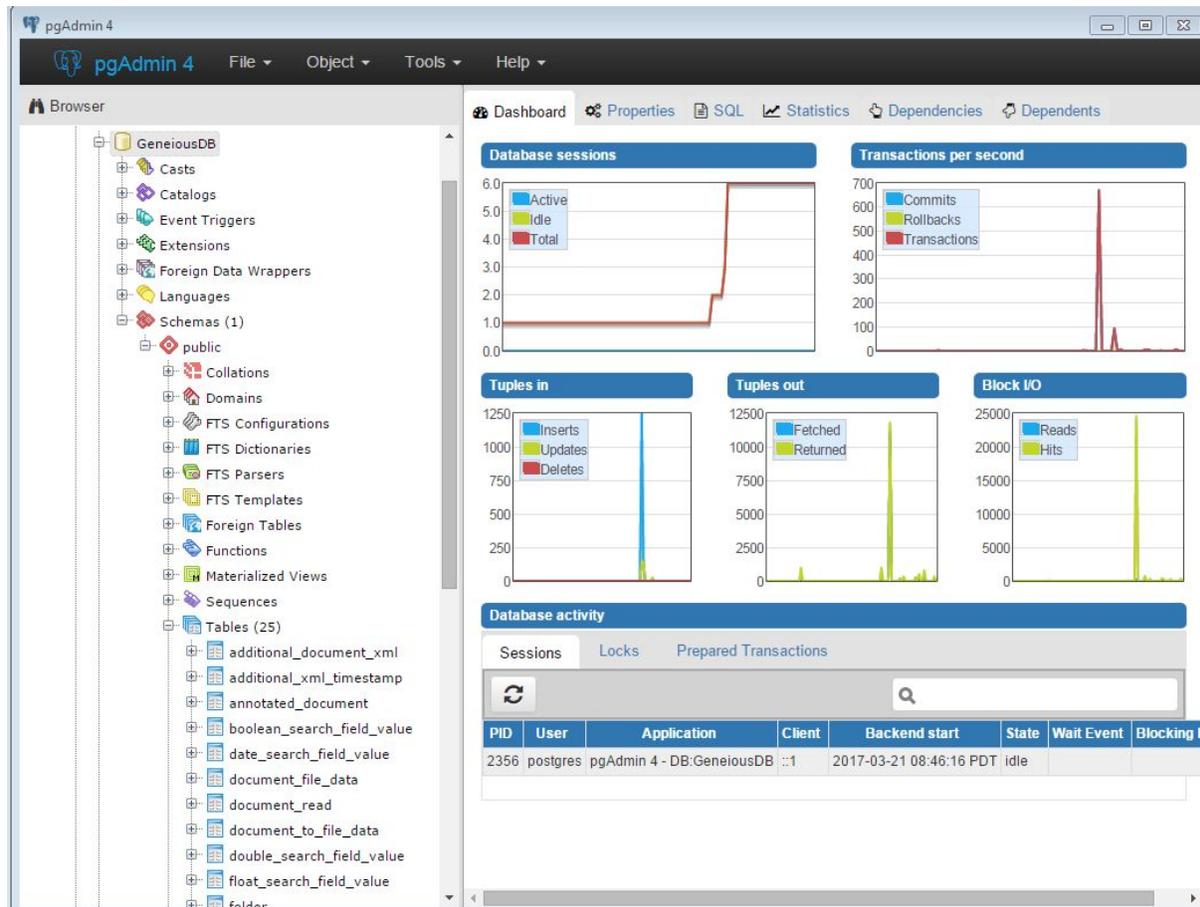
Enter the username and password for the superuser account, the IP address for the machine hosting the database (use localhost if it is on the same machine), and the database name. Geneious will then automatically finish the database setup by creating all the required tables.



A new folder called GeneiousDB will appear under your Shared Databases in the Sources panel. You can now start adding documents and creating new folders *via* Geneious.



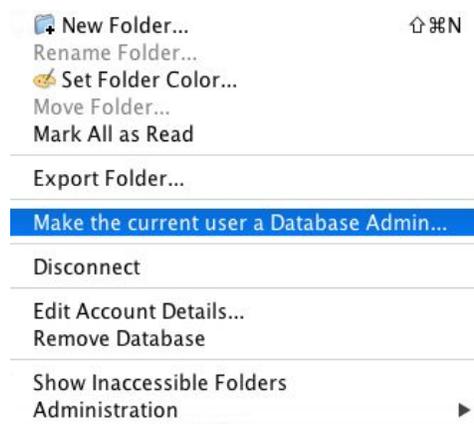
When you switch back to pgAdmin, you should now see that the database contains 25 tables under Schemas → public → Tables.



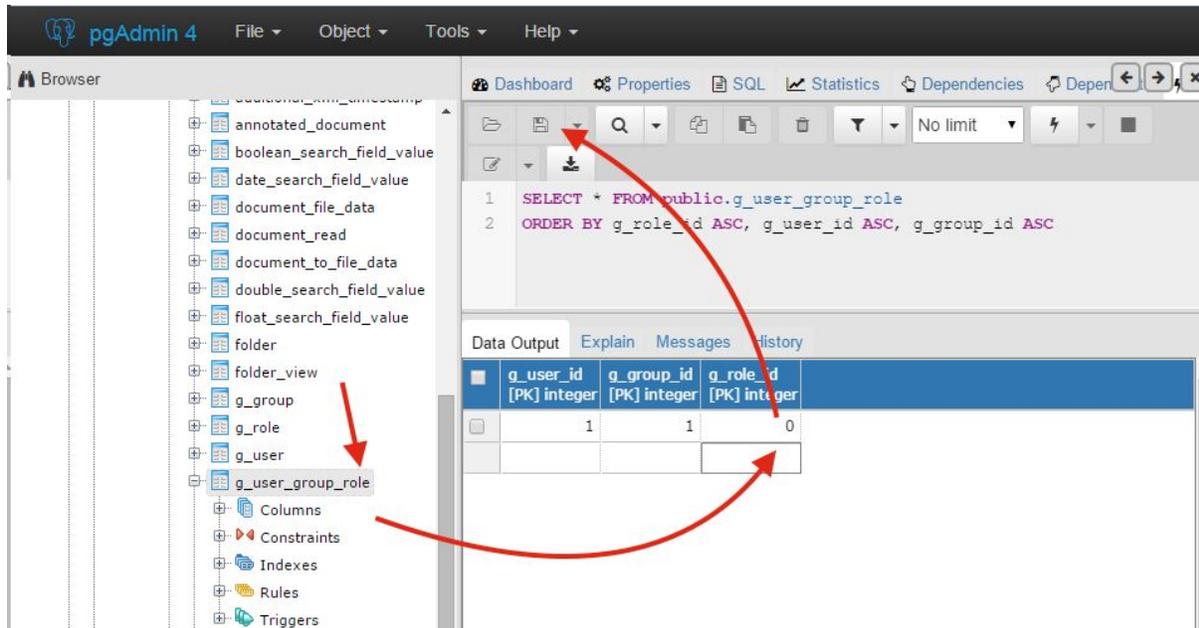
3. Creating an admin user

Being a superuser in PgAdmin does not automatically grant you administrator rights to the database in Geneious. There are two ways to grant administrator rights:

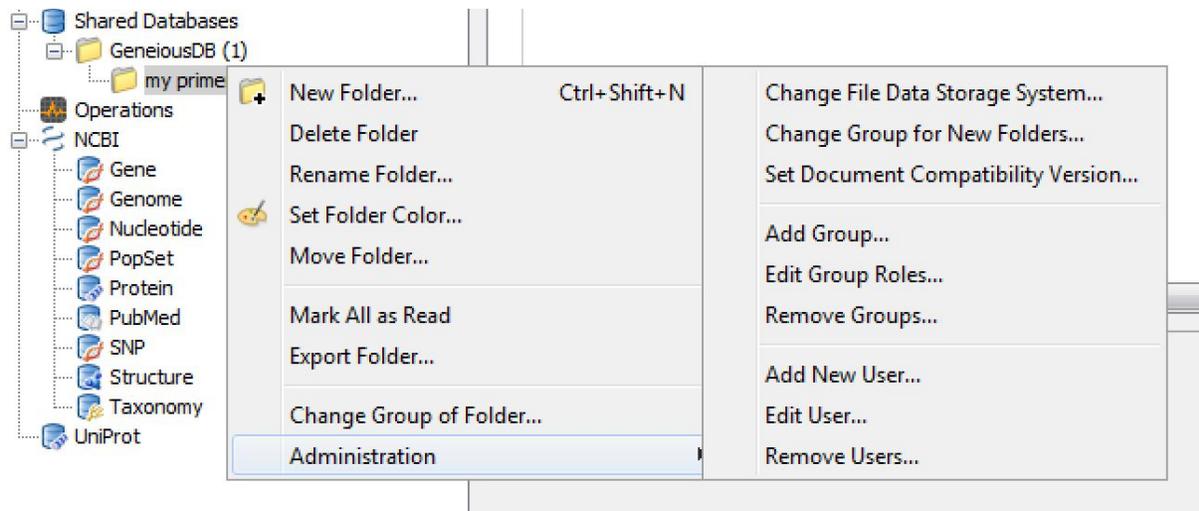
1. If this is a new database, and no one is currently an administrator, any user who is logged into the database in Geneious can set themselves as the database admin by right-clicking on the root folder of the database and choosing “make current user a Database Administrator”.



2. To add additional database admins, you need to edit the relevant tables in pgAdmin. Under Schema->Public->Tables find the table g_user. To view the table you need to right click on the table and select View/Edit data → View all rows. Check the ID (PK) Integer value for the user that you want to make an administrator. The owner of the database should be assigned the integer 1. Then go to the g_user_group_role table and view the contents. Find the g_user ID integer for the user that you want to make an administrator and modify g_role_id for that row to 0 by double clicking on the cell. The 0 allows for admin roles (to check the integers for other roles, check the g_role table). Remember to save (see below).



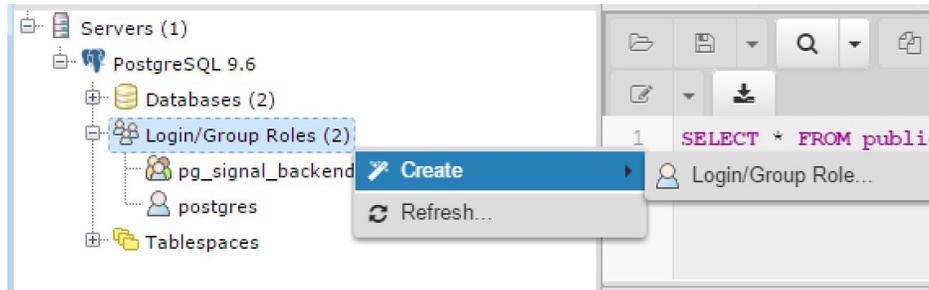
Now if you go back to Geneious and right click on any of your folders in your database, you will have access to the administration menu where you can perform functions such as defining groups and roles, and setting the Geneious compatibility version.



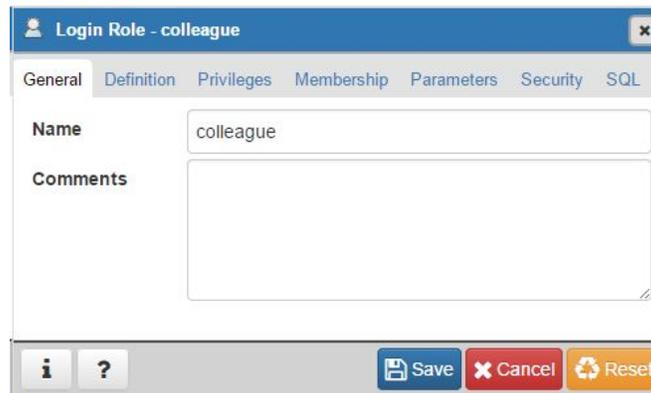
4. Add a new user

You will need to create new users in PgAdmin4.

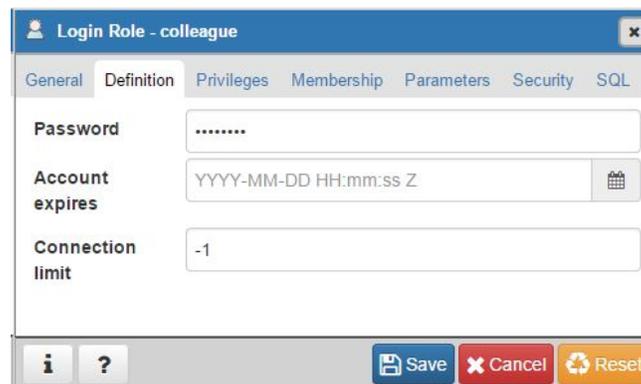
Open PgAdmin4, Select Login/Group Roles, right click and press Create → Login/Group Role



Add a name under the General tab:



Add a password under the definition tab:



Set the privileges as follows - this allows the user to log in to the database, but does not make them a superuser for postgres or grant them the ability to make new databases.

The screenshot shows the 'Create - Login/Group Role' dialog box in pgAdmin 4. The 'Privileges' tab is selected, and the 'SQL' sub-tab is active. The following table summarizes the privilege settings:

Privilege	Setting
Can login?	Yes
Superuser	No
Create roles?	No
Create databases?	No
Update catalog?	No
Inherit rights from the parent roles?	Yes
Can initiate streaming replication and backups?	No

You then need to grant the user SELECT, UPDATE, INSERT and DELETE permissions for all databases in the database so that they can use it. To do this, right click on the root Database folder in pgAdmin 4 and select "Grant Wizard". See [this link](#) for further information on the Grant Wizard.

In the first window, select all objects in the list to select all the tables.

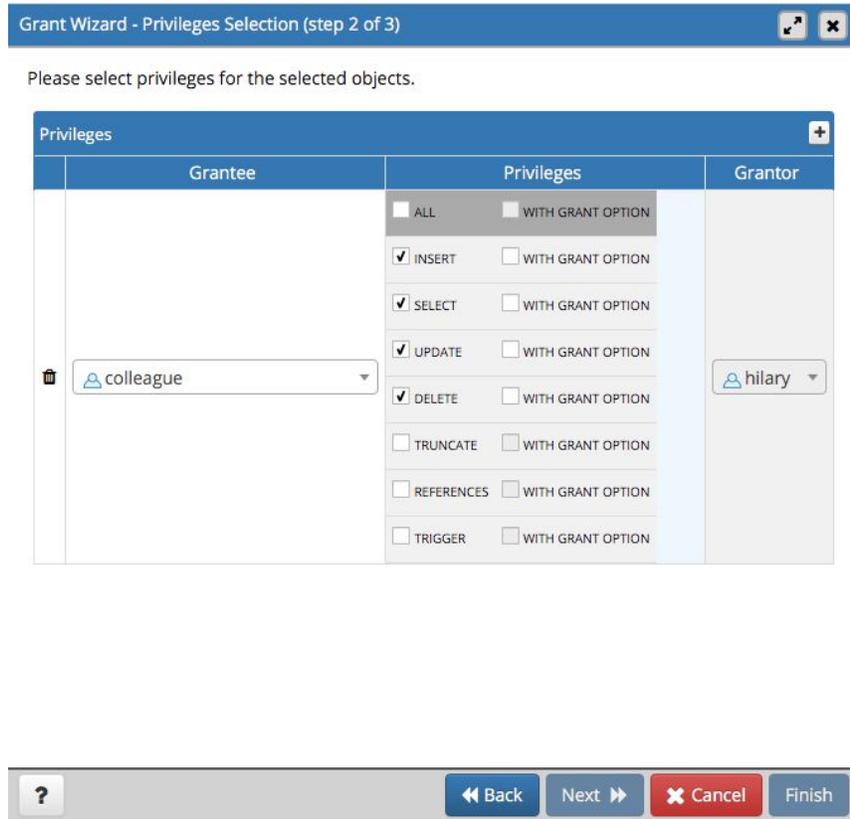
Grant Wizard - Object Selection (step 1 of 3)

Please select objects from the list below.

Search by object type or name

<input checked="" type="checkbox"/>	Object Type	Schema	Name
<input checked="" type="checkbox"/>	Table	public	additional_document_xml
<input checked="" type="checkbox"/>	Table	public	additional_xml_timestamp
<input checked="" type="checkbox"/>	Table	public	annotated_document
<input checked="" type="checkbox"/>	Table	public	boolean_search_field_value
<input checked="" type="checkbox"/>	Table	public	date_search_field_value
<input checked="" type="checkbox"/>	Table	public	document_file_data
<input checked="" type="checkbox"/>	Table	public	document_read
<input checked="" type="checkbox"/>	Table	public	document_to_file_data
<input checked="" type="checkbox"/>	Table	public	double_search_field_value
<input checked="" type="checkbox"/>	Table	public	float_search_field_value
<input checked="" type="checkbox"/>	Table	public	folder
<input checked="" type="checkbox"/>	Table	public	folder_view

Then click **Next**, and in the second window, click the **+** sign to grant new privileges. Select the user you just created under Grantee, then click Privileges and check Insert, Select, Update, and Delete. Click Next again to review the SQL command, and then **Finish**.

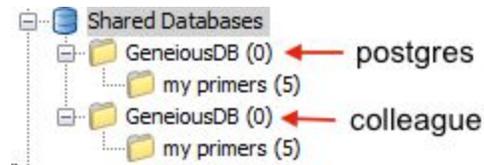


The user should now log into the database in Geneious by right clicking on the Shared Databases icon, and choosing “New database connection”. They should enter the username and password they were set up with in PgAdmin.

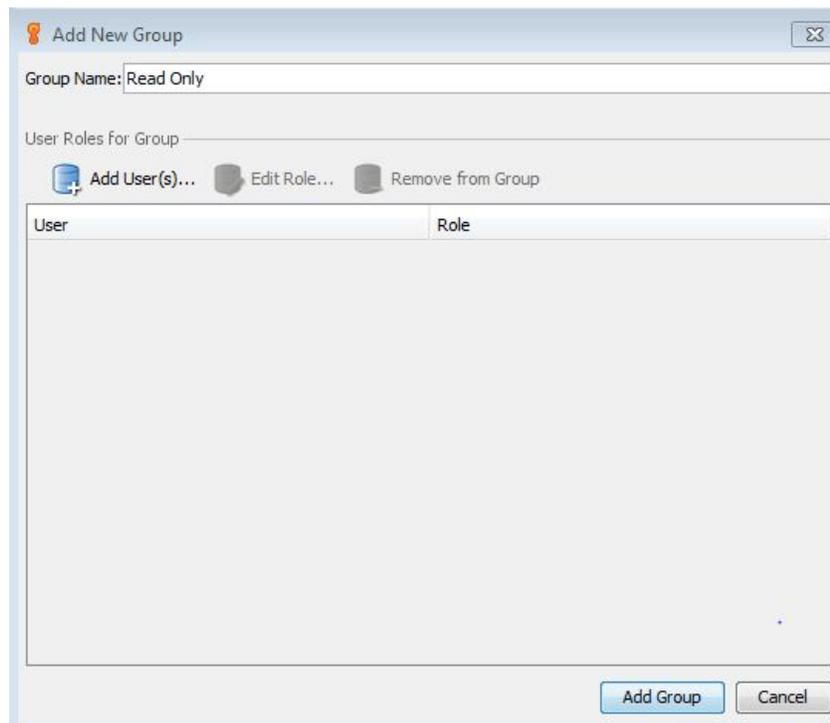
5. Changing Groups and Roles

For changing privileges and roles, you can log in as administrator in Geneious and use the Administration functions. By default some group roles will be automatically added, but you can create a new group, for example “Read Only”, and assign specific roles to each user. Options for roles are VIEW, EDIT and ADMIN.

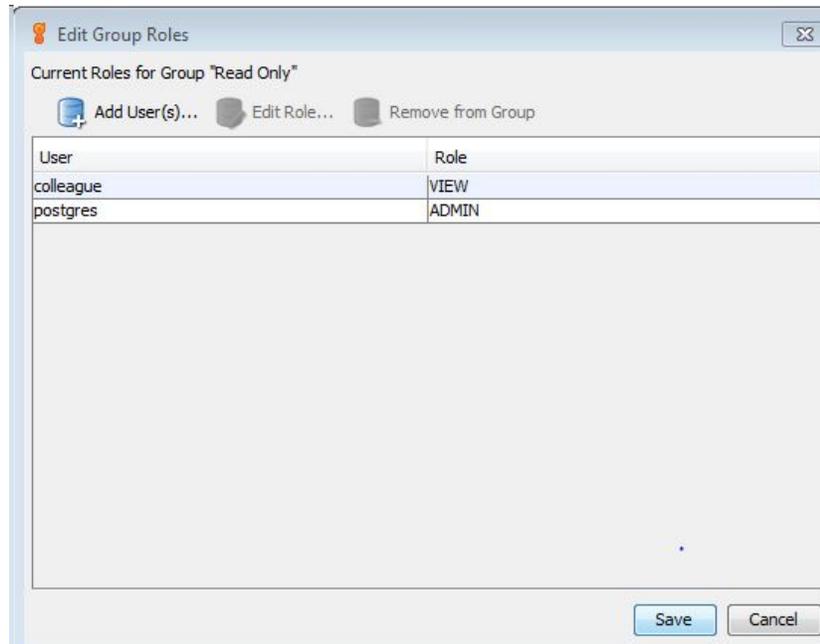
At the top, I’m logged in as postgres (superuser with administrator privileges). At the bottom I’m logged is as “colleague”, without administration privileges.



Right click on GeneiousDB at top and create a new folder. Call it “test”. It will also appear under the GeneiousDB user colleague. If you want to make this folder only visible to your colleague, with you retaining the right to modify the content of this folder, right click on any folder and select Administration → Add group. We can call the new group “Read Only”:



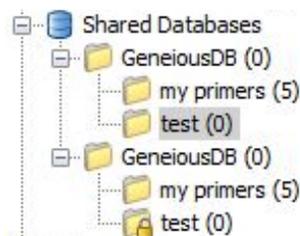
Click Add Group. Then right click again within the top postgres folders and select Administration → Edit Group Roles and select the new group “Read Only”. Here you can add the users that will be part of the group and associated to them roles, i.e. if they will be able to simply view the files (VIEW), to edit them (EDIT) or if they will have administration privileges (ADMIN). Let’s add the colleague with VIEW role and postgres with ADMIN role and save:



Now right click on the newly created folder “test” and select Change Group of Folder. Change it to the one we just created, “Read Only”.



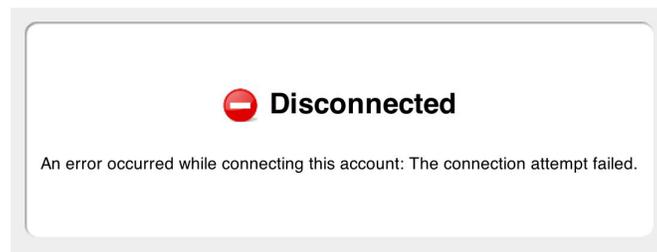
You will see that the folder “test” will become read-only and a lock will appear on the folder when logged in as colleague, with only VIEW rights:



6. Enabling remote connections to the database.

By default, PostgreSQL is not set to allow access by users from other computers on your local network.

If you try to connect from Geneious you will get the error message:



To enable access the following steps must be taken.

Step One: Make the Server Accessible

1. You should ensure the computer hosting your server has a fixed LAN IP address (This is usually configurable in your router).
2. On Windows: make sure that there is no firewall on your PostgreSQL server machine or that it has been configured to allow connections through port 5432

You can verify that PostgreSQL is allowed through the firewall on Windows by:

1. Opening the Control Panel
2. Selecting "Allow an app or feature through Windows Firewall"
 - This will be under either System & Security → Windows Firewall or, Windows Firewall depending on your settings
3. Finding the item in the table named "PostgreSQL". If it does not exist this means PostgreSQL is not allowed through the firewall
4. Clicking the "Details" button and verifying in the dialog that pops up that the "Port number" is 5432 and "TCP" is selected

PostgreSQL can be allowed through the firewall on Windows if you are an administrator by performing the following steps:

1. Open the table as per steps 1 and 2 above but choose "Advanced settings" instead of "Allow an app or feature through Windows Firewall" (this requires administrator privileges)

2. Click “Inbound Rules” and then “New Rule...”
3. In the dialog that opens select “Port” then work through the wizard making sure to use TCP and the local port 5432

If you are not able to make these changes you will have to contact your system administrator.

Step Two: Configuring PostgreSQL

Next you will have to edit the **pg_hba.conf** file. This file controls what connections the PostgreSQL database server on your computer will allow from other computers.

Follow the section corresponding to your Operating System below to open the file for editing:

Mac OS

1. In the Finder navigate to the PostgreSQL install, usually `/Library/PostgreSQL/X.X` (where X.X is the version you have installed)
2. Select the folder `data` and go **Get Info**. Click on the **padlock** in the bottom right corner to authenticate, then click on the **+** to add yourself with read/write privileges.
3. Open the data folder, select the `pg_hba.conf` file, **Get Info**, and again, authenticate, then click on the **+** to give yourself read/write privileges for this file.

Windows

1. Open up the File Explorer and navigate to `C:\Program Files\PostgreSQL\9.6\data`
2. Open up `pg_hba.conf` in your favourite text editor such as WordPad.

Edits to Make

In the Configuration table add the line:

```
host    all    all    xxx.xxx.xxx.0/yy    md5
```

For example if your local network assigns IP addresses in the format of 192.168.x.y then use:

```
host    all    all    192.168.0.0/16    md5
```

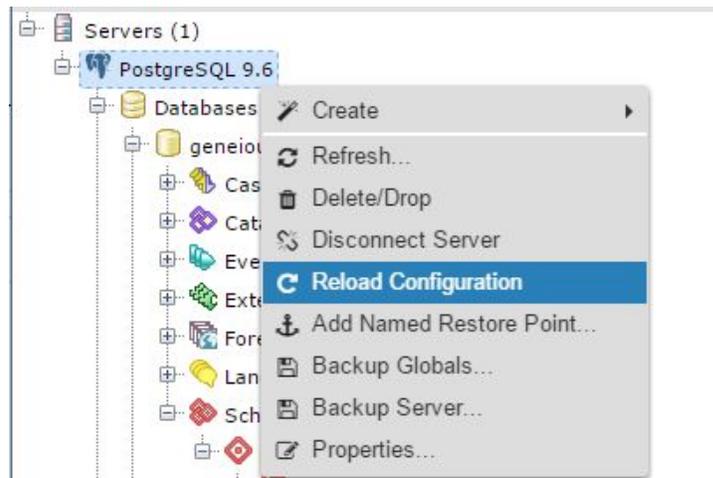
If you want to use SSL in this case then use the line:

```
hostssl all    all    192.168.0.0/16    md5
```

For further information on configuring `pg_hba.conf` see <https://www.postgresql.org/docs/current/static/auth-pg-hba-conf.html>

Restart the Server

Once you have saved your edits, restart the server. In pgAdmin4 you can right click on the server node and choose Reload Configuration.



You should now be able to connect to your posgreSQL database from another computer provided you have valid login/password credentials set up.

