

Creating Your Own Database Using PostgreSQL

Introduction	3
Administrator Rights	3
Rationale	3
Installation	5
Download PostgreSQL Installer	5
Install PostgreSQL	6
Installation Verification	13
Using psql	13
Using pgAdmin	14
Creating a database	15
Creating a Database using pgAdmin	16
Creating Tables in pgAdmin	20
Importing Data into your Database	21
Permission Denied	22
Summary	22

Introduction

This is a guide to help you build your very own database using a PostgreSQL server. It will show you how to install a PostgreSQL server, create a database, create tables and import data from a csv file into your database.

Administrator Rights

You will need administrator rights on the device you are planning to use for this tutorial

If you plan to use a personal device then this should be fine

If you plan to use a work device please ensure you have followed the correct procedure relevant to your organisation and have received the correct authorisation to do so

Rationale

This will build upon your general knowledge regarding database management systems (module 2) and SQL (bootcamp)

The subsequent use of your database could potentially allow you to show evidence for the four competencies listed in Table 1 on the next page. However this is dependent on whether you can:

- Use **work relevant** data on a personal device
 - BCS will allow you to use open source data to do some analysis as long as you can justify to the assessor in your interview and portfolio why you chose to use it.
 - You can use your own devices to analyse **work relevant** data.
 - **Work relevant** data is harder to justify than using work data.
- Use **work data** on a work device
 - Obviously it is easy to justify the use of **work data** but you should only analyse **work data** on work devices to ensure you are not violating any internal or external data security and/or sharing regulations.
 - You need permission from work to install a DBMS on a work device and to ensure you have the correct permissions to import data from your devices' drive - this may not be possible for many of you.

Key	Competency
TC2	Manipulate and link different data sets as required
TC3	Interpret and apply the organisations data and information security standards, policies and procedures to data management activities
TC4	Collect and compile data from different sources
TC5	Perform database queries across multiple tables to extract data for analysis

Table 1: Technical Competencies

We are going to install a PostgreSQL DBMS; the user interface (pgadmin) is something you are familiar with from bootcamp and the syntax is identical to that which you have learnt already. Not only that but it is a really cool open source database solution that allows you to do pretty much whatever you want with it.

PostgreSQL is an Object-Relational Database Management System (ORDBMS) so slightly different to those covered in module 2 but basically the same as a RDBMS but more efficient.

NB

If you want to aim for a distinction you have to show that you have gone *above and beyond* in how you gathered evidence for a number of technical competencies. Obviously designing and building databases is a data engineers' role and not a task expected from most data analysts. However if you do design and build your own database and use it to run some queries that are discussed in a portfolio item (project or reflective journal) this will provide good evidence that you went *above and beyond* the remit of your role. On its own it won't be enough to secure a distinction, you'll most likely have to show you've gone *above and beyond* for other competencies as well to secure an overall distinction but this is one area that is relatively straightforward for you to achieve compared to other competencies like TC7.

Installation

This guide is for installing PostgreSQL onto a Windows device, please use the following link if you want to install PostgreSQL on a [Mac](#)

Download PostgreSQL Installer

Use the link [here](#) to go to the PostgreSQL database download page as shown in Figure 1 below. Click on the download link as required, latest version as of Feb 2021 is 13.1 and is recommended. It will take a few minutes to complete the download.

Version	Linux x86-64	Linux x86-32	Mac OS X	Windows x86-64	Windows x86-32
13.1	N/A	N/A	Download	Download	N/A
12.5	N/A	N/A	Download	Download	N/A
11.10	N/A	N/A	Download	Download	N/A
10.15	Download				
9.6.20	Download				
9.5.24	Download				
9.4.26 (Not Supported)	Download				
9.3.25 (Not Supported)	Download				

Figure 1: PostgreSQL download versions

Use the link [here](#) to find out more about x32 and x64 and then use this link [here](#) to find out how you can tell if you have a x32 or x64 device for Windows

Install PostgreSQL

PostgreSQL has an easy to use setup Wizard that will take you through the installation process. For most of us, we can just accept the default options, however if you want to create a more complex database you may want to specify some other options. These other options are beyond the scope of this document.

Step 1 - Run the executable file (.exe)

In your download folder, double click on the installer file - *postgresql-13.1-1-windows-x64.exe*.

Step 2 - Enter Administrative Credentials

Enter your administrator username and password.

Step 3 - Start the Setup Wizard

The installation Wizard will appear and guide you through the next few steps. Click on the *next* button to get started as shown in Figure 2 below.

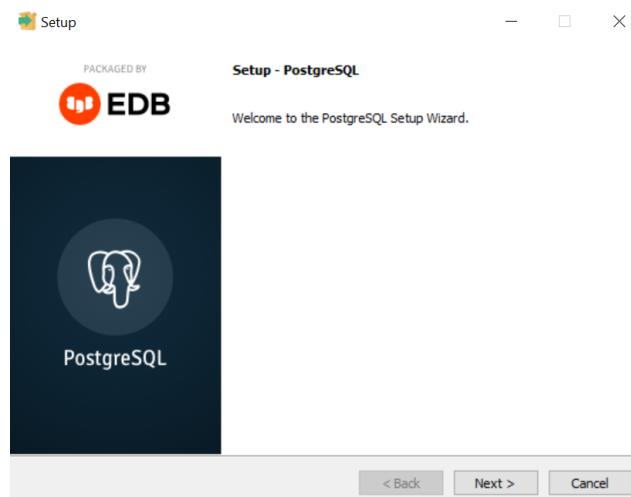


Figure 2: Setup Wizard start

Step 4 - Specify the Installation Folder

Specify the folder you want PostgreSQL to be installed in - for most purposes it is fine in the default folder suggested by the set-up Wizard - to accept the default click on the next button shown in Figure 3 on the next page.

If you want to install it elsewhere, navigate to the folder you want and click the *next* button.

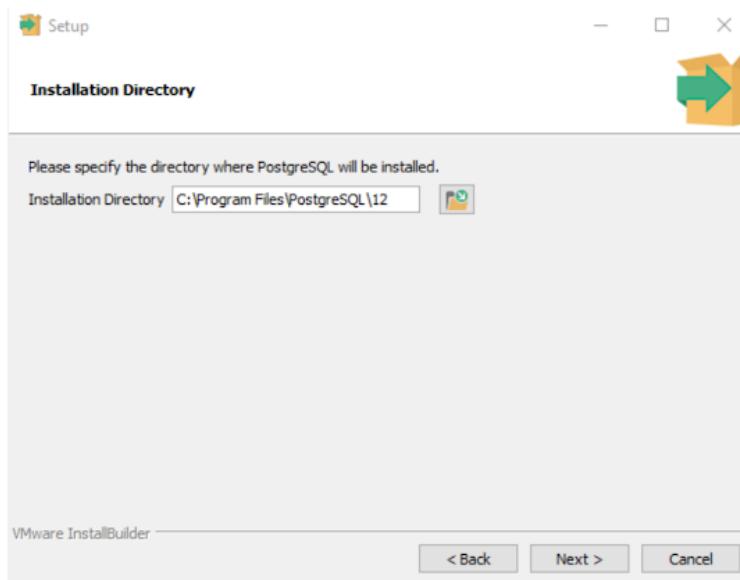


Figure 3: Choose your installation folder

Step 5 - Select Software Components

Select the software components to install - see Figure 4 overleaf for more detail. Then click *next* when done.

1. Ensure the **PostgreSQL Server** box is checked - this installs the server.
A server is a computer connected to a network of other computers called clients. Client computers request information from servers over the network. Servers tend to have more storage, memory and processing power than a client computer.
2. Ensure **pgAdmin 4** box is checked - this installs the Graphic User Interface (GUI) tool you got to know and love so much in bootcamp.
3. Ensure **Command Line Tools** is checked - this installs **psql** and other tools that allow you to interact with the PostgreSQL database server using the Command Line Interface (CLI).

4. Optional - **Stack Builder** - this installs a GUI that will allow you to install other drivers that interact with PostgreSQL. This is beyond the scope of what we need a database for so I left mine unchecked.

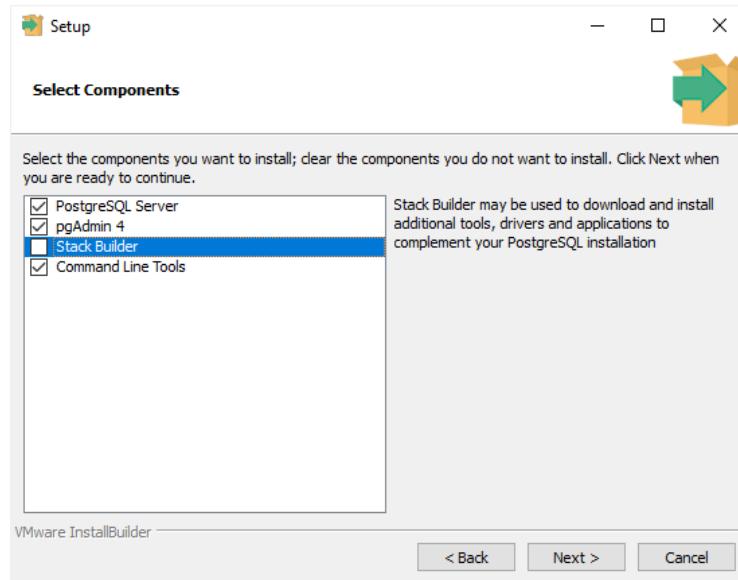


Figure 4: Software Components

Step 6 - Select database directory

Select the database directory you want to store the data in or accept the default location as shown in Figure 5 below. Once you've selected your chosen directory, click on the *next* button when done.

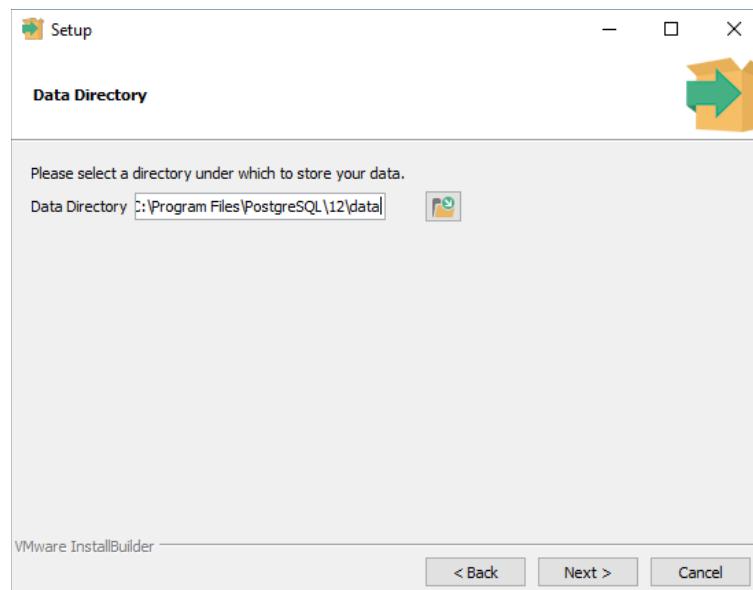


Figure 5: Select directory for the database

Step 7 - Database Password

Enter the password for the database superuser - postgres (you) as shown in Figure 6. Retype the password again to confirm. Click *next* when done.

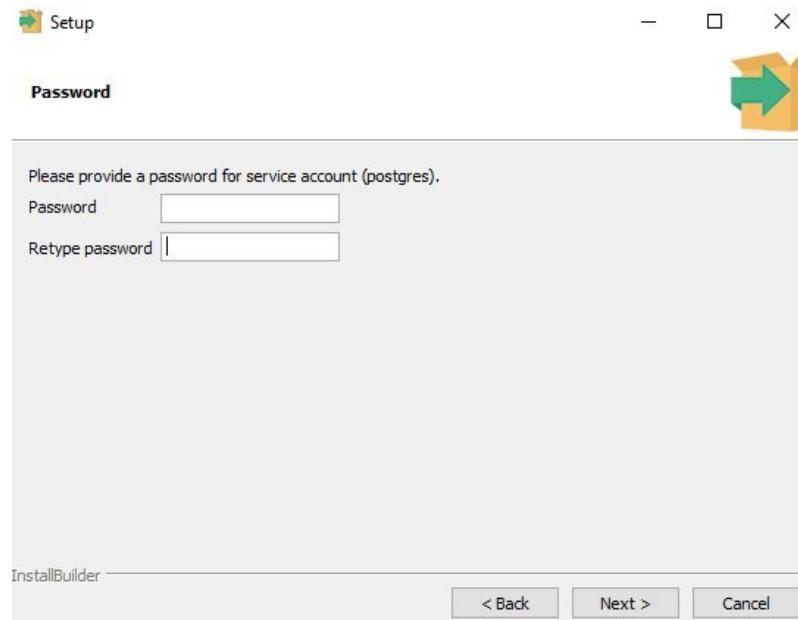


Fig 6: Enter your database password

Step 8 - Port Number

Enter the port number (as shown in Figure 7 on the next page) on which the PostgreSQL database server will listen - the default option is 5432.

You will only need to change this if you have another application using the same port. That means every time you request something from the server, it will receive this request through port 5432. Once you have selected your port, click on the *next* button.

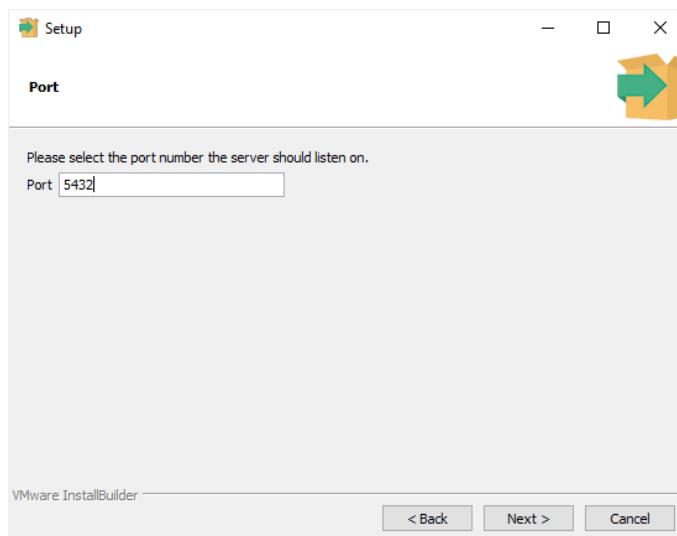


Figure 7: Enter port number

Step 9 - Select Locale

Choose the default locale as shown in Figure 8. In computing a *locale* is a set of parameters that defines the user's language, region and other preferences. By choosing the default option you will ensure that the system is set up for American English (remember things like colour is color, etc). Once you have selected your locale, click on the *next* button.

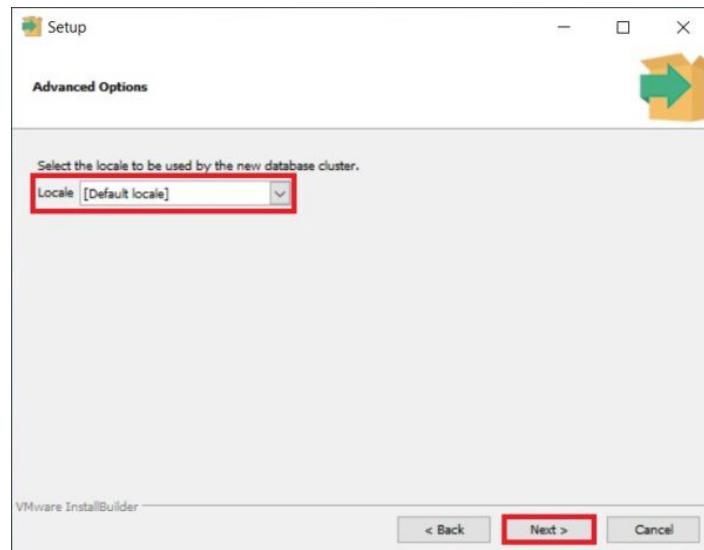


Figure 8: Selecting your locale

Step 10 - Setup Wizard Summary

The setup Wizard will show the summary information of your PostgreSQL installation as shown in Figure 9 below. Review it and if happy with the summary – click the *Next* button.

If unhappy with the summary, click the *Back* button to change the configuration accordingly.

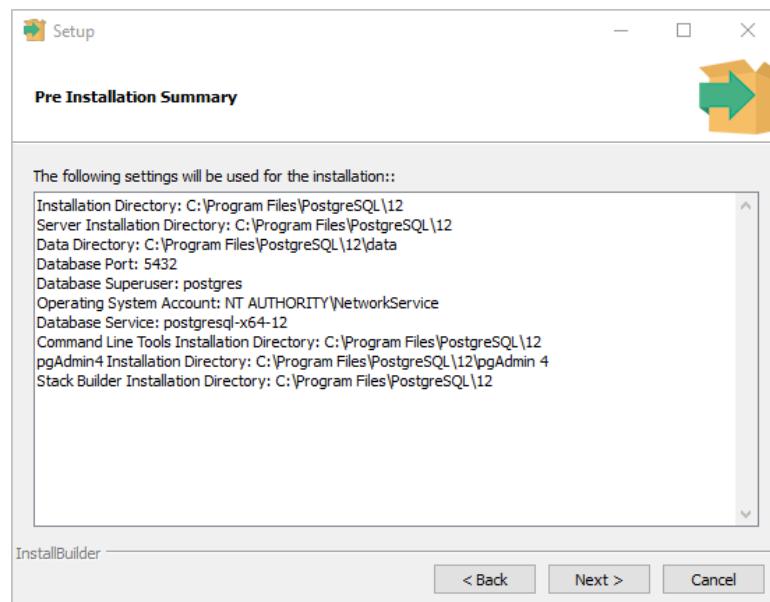


Figure 9: Setup summary

Step 11 - Install

Once you're happy with the setup summary, click *Next* to begin installing PostgreSQL as shown in Figure 10 overleaf.

The installation may take a few minutes to complete.

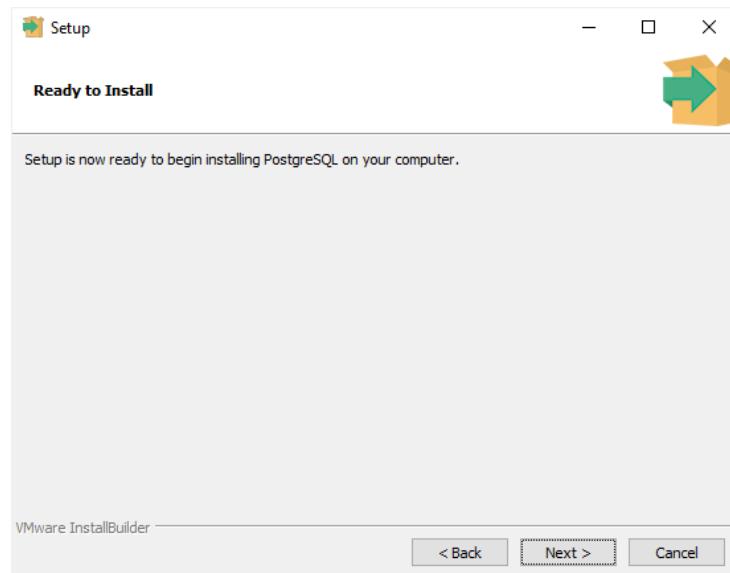


Figure 10: Pre-installation stage

Step 12- Finish Installation

Once done, click on the *Finish* button to complete the installation as shown in Figure 11 below.

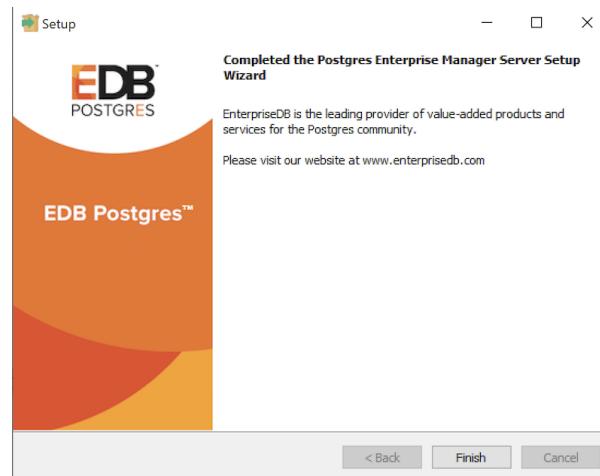


Figure 11: Finish installation

Installation Verification

There are two methods for checking you have installed the software correctly.

Using psql

In your start menu, type 'psql' and then select the psql application when it pops up.

When the command prompt pops up press *Enter* four times. This means you are accepting the default server, database, port and username respectively.

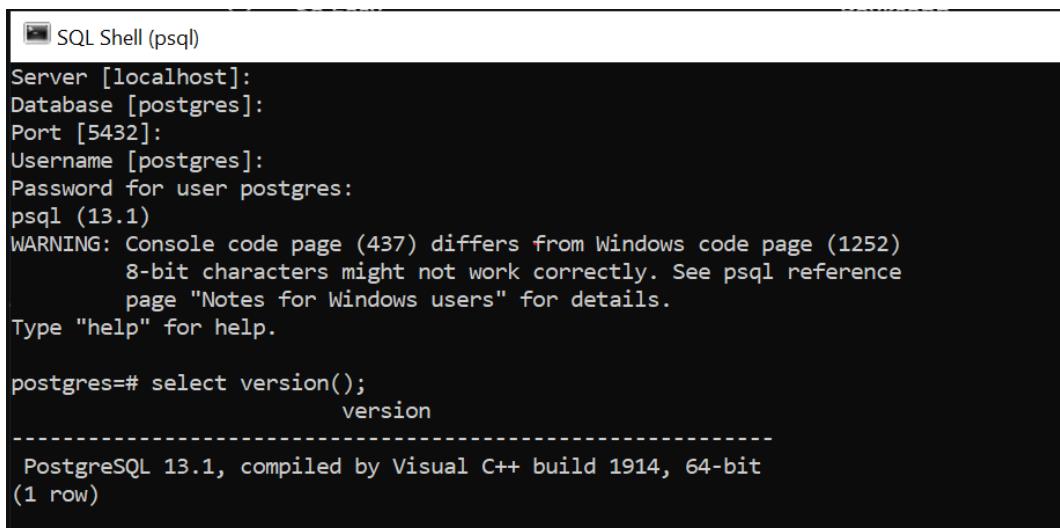
Then enter your password. This is the one you entered into the setup wizard in stage 7 of the installation process.

A warning will pop up (lines 7-10 of Figure 12), ignore that and type the following command,

```
select version();
```

Be sure to use the semicolon at the end of the select command above. Then press *Enter*.

The output that follows (Figure 12 below) states that this version is version 13.1 which is the one downloaded from the Postgres download page as shown in Figure 1. This message lets you know you have installed the database server on your local device correctly



```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Password for user postgres:
psql (13.1)
WARNING: Console code page (437) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

postgres=# select version();
              version
-----
 PostgreSQL 13.1, compiled by Visual C++ build 1914, 64-bit
(1 row)
```

Figure 12: Installation verification using psql

Using pgAdmin

In your start menu, type pgAdmin and then select the application.

Once pgAdmin is up and running, click *Servers* in the navigation bar on the left hand-side. You will be prompted to enter the password you set up in stage 7 of the installation process.

Once your password has been entered correctly, select *Properties* in the grey banner at the top of the main pane as shown in Figure 13 below.

In the *General* box, next to *Version*, you will see your current version, in Figure 13 this is version 13.1.

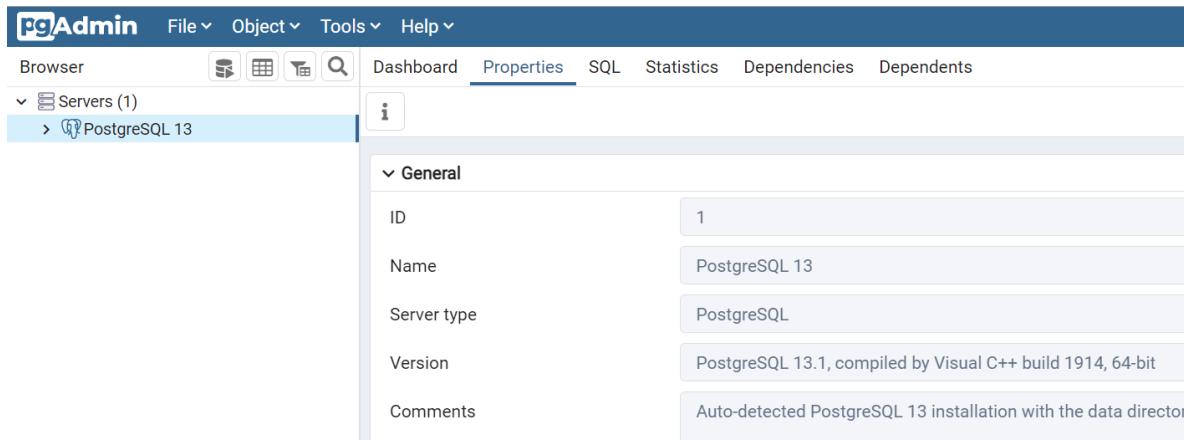


Figure 13: Installation verification using pgadmin

Now you have successfully installed PostgreSQL, let's learn how to create tables and upload csv files.

Creating a database

This guide is not exhaustive, there are many more commands that are not covered such as UPDATE, ALTER, DROP, etc that are incredibly useful to understand when building and maintaining a database.

The following guide assumes you have cleaned your data sets that you want to upload - the tables are normalised to 3NF (or BCNF or ONF), there are strategies in place for nulls and missing data, consistent formats (dates, times, names, etc), all the good stuff from modules 1 and 2 as well as the data analytics bootcamp.

For a project involving the use of your database, it would be a good idea to include the following in your write-up:

- The data sources.
- The advantage of using your own database.
- How have you identified, collected and cleaned the data?
- How have you normalised your data?
- Consider including an ERD, conceptual/logical/physical models - link to [lucidcharts](#).
- The steps you took to create your database, tables and importing the data.
- Any internal and/or external regulations you had to adhere to in order to use the data you have used?
- Why did you run the queries you ran?
- If work relevant data - what is your **justification** for using the data you have used ?
- Any limitations of using PostgreSQL or your database design?
- Lessons learnt - what went well, what went not so well, what would you do differently if you did it again, etc?
- Remember to include, BCS Terminology such structured/unstructured data, entities, relations, fields, records, etc.

Creating a Database using pgAdmin

pgAdmin is a tool that you should be familiar with having used a similar version in bootcamp.

Open up pgAdmin 4 (either through the start menu or desktop shortcut if applicable).

You will be prompted to enter your password from Stage 7 of the installation process. Once entered correctly you will see the familiar graphic user interface (GUI).

In the left hand navigation pane click on *Servers*. You will be prompted to provide your password again. Once entered correctly navigate to *Databases (1)*.

Servers > PostgreSQL 13 > Databases (1)

Right clicking on *Databases (1)* will provide you with the option of creating a database (Figure 14) using the Database Dialog box.

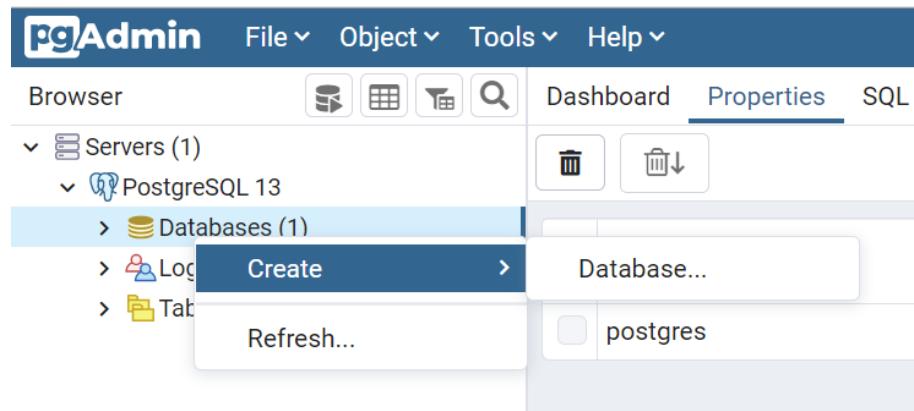


Figure 14: Right click option to create database

There are a number of options in the [Database Dialog box](#) (Figure 15) and it's worth investigating these further.

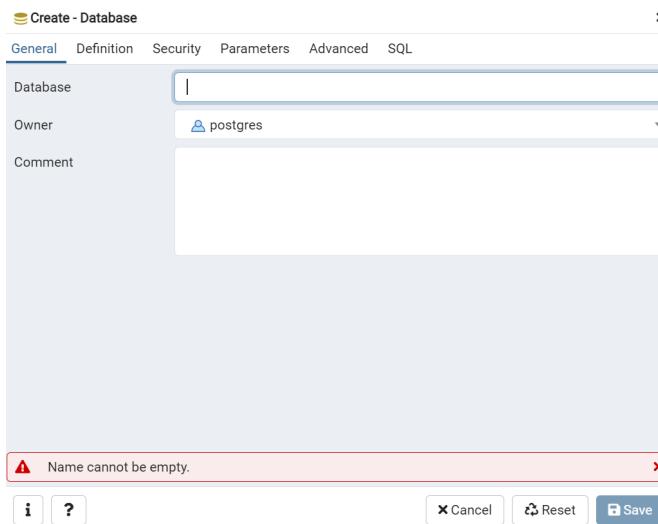


Figure 15: Database Dialog box

However, that's relatively straightforward and doing this doesn't let you show off your SQL skills. So we're going to create one using the `CREATE DATABASE` statement.

First, navigate through the *Databases (1)* structure until *postgres*.

Servers > PostgreSQL 13 > Databases (1) > postgres

Click on *postgres* to connect to the server. The previously greyed-out *Query Tool* (Figure 16) icon will now become active, click on it to open the *Query Editor*.

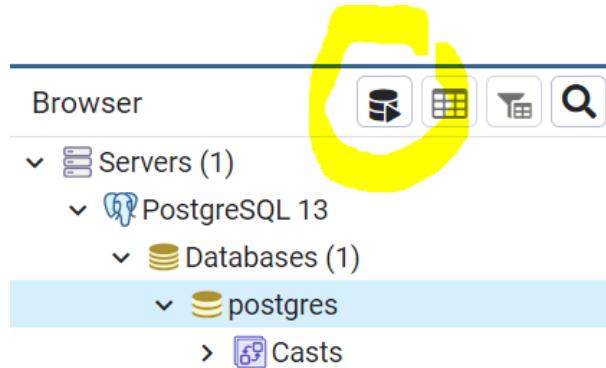


Figure 16: Query Tool Icon highlighted in yellow

Figure 17 shows a CREATE DATABASE statement.

```
CREATE DATABASE happiness
    WITH
        OWNER = postgres
        ENCODING = 'UTF8'
        CONNECTION LIMIT = -1;
```

Figure 17: CREATE DATABASE statement

To understand this statement fully, let's break it down and go through it line by line.

CREATE DATABASE happiness

Unsurprisingly, this line creates a database called *happiness*.

WITH

This is a clause that allows you to write auxiliary statements for use in larger or more complicated queries. It is also necessary here to specify certain database properties (such as owner, encoding and connection limit) when creating the database. Click [here](#) to find out more about the WITH clause.

OWNER = postgres

When you create a database you become its owner. I created this database so my username is `postgres`, hence `postgres` owns this database.

In order to allow other users to use your database, you can grant *privileges*. If you want to find more about the different types of privileges head to the [PostgreSQL documentation](#) website.

Encoding = 'UTF8'

Encoding keeps data safe and file size small, since encoded data is smaller in size you can store more of it on your device. This makes the underlying software and hardware components of your device run more efficiently.

UTF8 is a common form of encoding. Notice it is entered as a *string* data type.

Connection Limit = -1

The connection limit configures the number of connection requests. **-1** is the default value and allows you maximum connections to the database.

A database connection specifies the necessary information for connecting to a specific database as a specific user of that database.

Run Query

After you have written the query, press *Run*. You will receive notification of successful database creation in the message panel as shown in Figure 18.

```
Data Output Explain Messages Notifications  
CREATE DATABASE  
Query returned successfully in 1 secs 550 msec.
```

Figure 18: Create database successful message

Refresh your web page to see the new database appear in the navigation panel as per Figure 19. The new database will be greyed out at first, click on it to connect to it (changes to the yellow colour as shown in Figure 19).

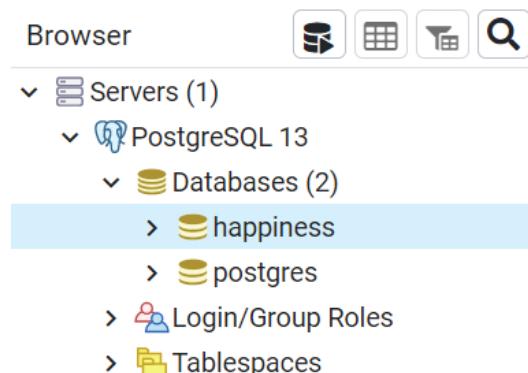


Figure 19: Newly created database

Creating Tables in pgAdmin

Now that you have created your database it's time to provide data structures (tables) using the CREATE TABLE function as shown in Figure 20 below.

```
CREATE TABLE global_happiness (
    country_name text,
    regional_indicator text,
    score float,
    life_expectancy float,
    PRIMARY KEY(country_name)
);
```

Figure 20: CREATE TABLE statement

CREATE TABLE global_happiness();

This creates a table (data structure) called *global_happiness*.

Within the parentheses we specify important properties of the table such as

- the column names, e.g., **country_name**
- their respective data types, e.g., **text**
- and the primary key, e.g., **PRIMARY KEY**(*country_name*)

Note how each column name is separated by a comma.

Ensure you close off your parentheses and complete the query with a semicolon.

Run Query

Press *Run*. A message notifying you that your table has been created successfully will appear in the bottom message pane.

Refresh your web page and then navigate to *Tables* in the navigation pane. Click on *Tables* and you will see your new table.

If you click on *Columns* you will see the names of the columns you specified in your CREATE TABLE function.

Importing Data into your Database

Now you have created your database and tables, the last thing to do is to import your data into the database. This is done by using the COPY statement shown in Figure 21 below.

```
COPY global_happiness (
    country_name,
    regional_indicator,
    score,
    life_expectancy
)
FROM 'C:\Users\████████\████\Database\world_happiness_clean.csv'
DELIMITER ','
CSV HEADER;
```

Figure 21: The COPY statement

COPY global_happiness ()

The COPY statement takes the name of the table that will hold the imported data.

In order to copy the data, a table must have been created with the proper structure (same number of columns, correct data types, etc).

FROM 'filepath'

Just like the select statements you did in bootcamp needed a FROM clause to point to the correct table name. The COPY statement needs the FROM clause to point to where you store the data you want to import.

If you copy and paste your filepath, the pasted quotation marks that wrap your filepath will not be recognised by pgAdmin. Replace the pasted double quotation marks with single ones as shown in Figure 21.

DELIMITER ','

CSV files usually have values separated by commas (however this is not always the case). When the data is being imported into the table it uses these commas to know where the columns begin and end.

CSV HEADER:

A header is a file which contains the column names as the first line of values in the file. Not all csv files have a header. If the file you are uploading has no header then there is no need to use this line in your COPY statement.

Import Verification

Once you have successfully run your COPY statement, you can check the data has been imported correctly by typing a simple SELECT * query and checking the number of columns and rows is as expected and the data types are correct for each column.

Once you are happy you have imported the data correctly, create more tables and import data into these other tables. Then you can join, union, aggregate data and nest queries to create your very own data sets to analyse.

Permission Denied

Depending on your device's administration privileges you may encounter an error message like that in Figure 22 below when trying to copy data from your device into your database.

```
ERROR: could not open file "C:\Users\████████\████████\████\████\gmass\gmass_demo.csv"
for reading: Permission denied
```

Figure 22: Permission Denied message when copying data from your device into your database

There are two methods to solve this, you can use i) psql or ii) you can alter the file properties.

1. Use psql

The COPY statement written in pgAdmin (Figure 21) creates a request from the database server to your device. Since this is an external request to your device and you haven't provided any privileges, this request is denied by your device.

psql has been installed on your device so writing the copy statement in psql (shown in Figure 23) means that the request to read the file is coming from your device and not an external source, therefore permission to read the file is granted.

In your start menu, type *psql* and double-click on it to open it up. Remember to accept the default first four options by pressing *Enter* four times in a row, then you will be prompted to enter your password.

Once you have entered your password, type the following to connect to the applicable database.

```
/c your_database_name
```

Once you're connected to your database you can go ahead and enter the COPY statement as shown in Figure 23.

```
happiness=# \copy gmass_demo from 'C:\Users\████████\████████\████\████\gmass\gmass_demo.csv'
with delimiter ',';
COPY 3
happiness=#
```

Figure 23: copy statement in psql

Be sure to refresh your pgAdmin interface to see the new table appear and then throw in a validating SELECT * statement to check the rows and columns are as expected.

2. File Properties

In your file explorer, navigate to the file you want to import using the copy statement. Right click on it and go to properties. Select the *Security* as shown in Figure 24.

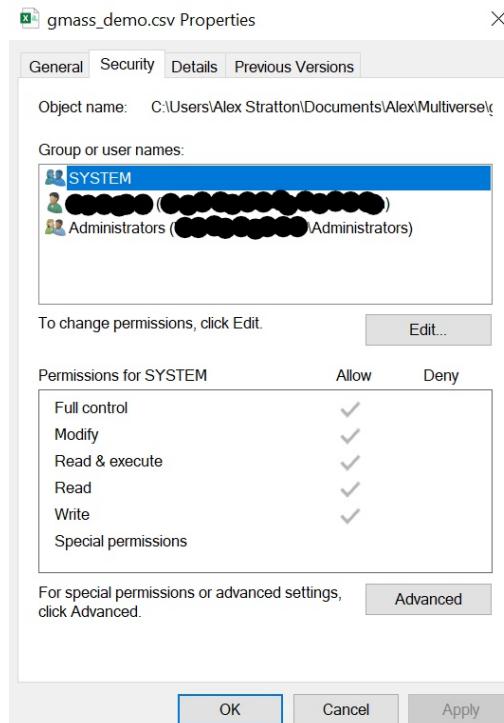


Figure 24: Properties Security tab

Click on *Edit*, then click on *Add*, this allows you to enter a new group (in Figure 25 this new group is called *Everyone*).

Ensure that you have selected the *Full Control* tick box in the *Permissions for Everyone* box as shown in the bottom half of Figure 25.

Press *Apply* then *OK* to exit the security tab. Then *OK* to exit out of the properties tab.

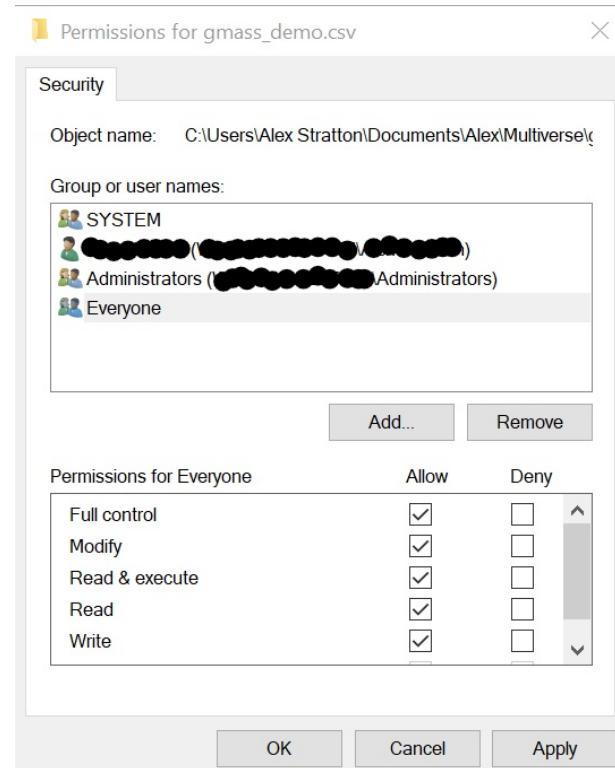


Figure 25: Creating a new group with full control

Now you can return to your pgAdmin interface and run the COPY statement again, this time the server will be able to read the file and you will be able to copy the data from your file and into your database table.

Summary

This guide has shown you how to:

- Create your own database
- Create your own tables
- Import your data into your database

There is so much more to do in terms of developing databases but the three points listed above are enough to get you querying your own databases and allowing you to enhance your SQL skills post bootcamp.