

# Installing & Using PostgreSQL

Updated: 14 August 2019

This guide explains how to install PostgreSQL, an open-source object-relational database management system on your laptop/desktop computer that you will be using for CS2102. It also describes how to use `psql`, PostgreSQL's command-line client program to access a running PostgreSQL server.

## 1. Installing PostgreSQL

In the following, we describe a convenient way to install PostgreSQL for three main OS platforms. Other options are described at <https://www.postgresql.org/download/>.

## Downloads

### PostgreSQL Core Distribution

---

The core of the PostgreSQL object-relational database management system is available in several source and binary formats.

### Binary packages

Pre-built binary packages are available for a number of different operating systems:

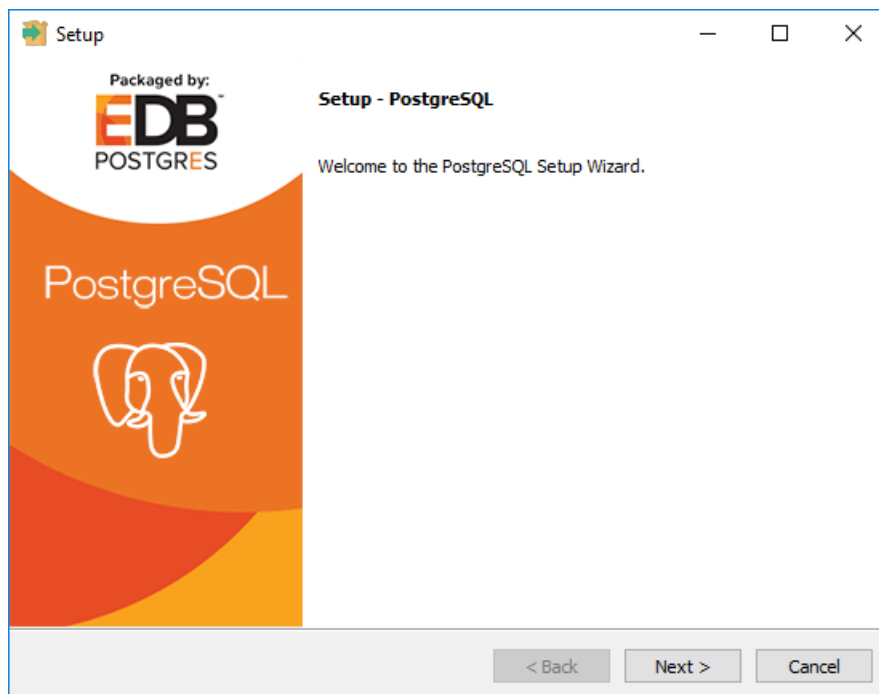
- BSD
  - **FreeBSD**
  - **OpenBSD**
- Linux
  - **Red Hat** family Linux (including **CentOS/Fedora/Scientific/Oracle** variants)
  - **Debian** GNU/Linux and derivatives
  - **Ubuntu** Linux and derivatives
  - **SuSE** and **OpenSuSE**
  - **Other** Linux
- **macOS**
- **Solaris**
- **Windows**

## 1.1. MacOS/Windows

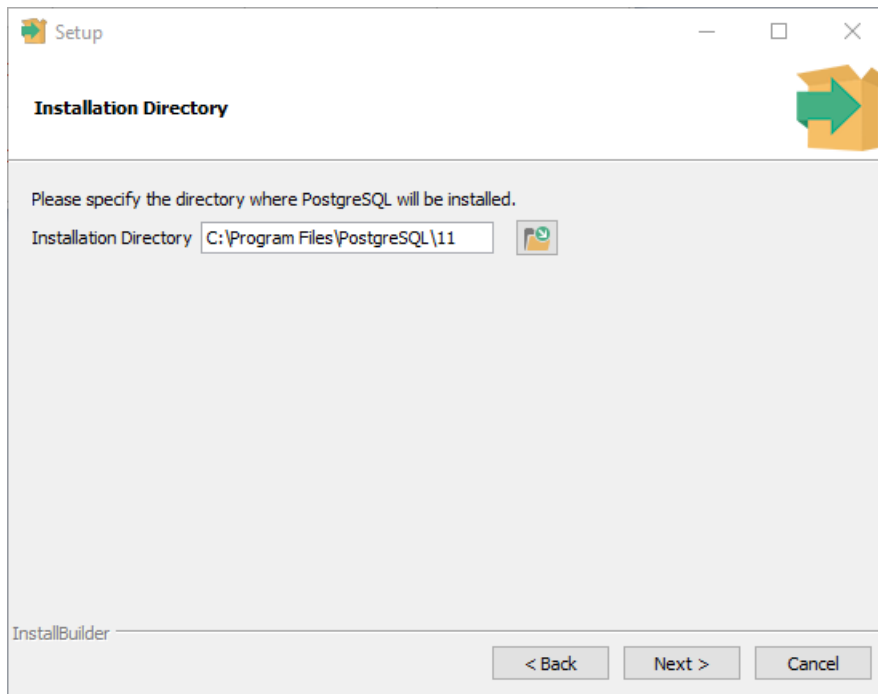
1. **Download** the EnterpriseDB installer for PostgreSQL 11.5 from <https://www.postgresql.org/download/windows/>.

PostgreSQL Version	Windows x86-64	Mac OS X	Linux x86-32	Linux x86-64	Windows x86-32
11.5	<a href="#">Download</a>	<a href="#">Download</a>	N/A	N/A	N/A
10.10	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>
9.6.15	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>
9.5.19	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>
9.4.24	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>
9.3.25 (Not Supported)	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>

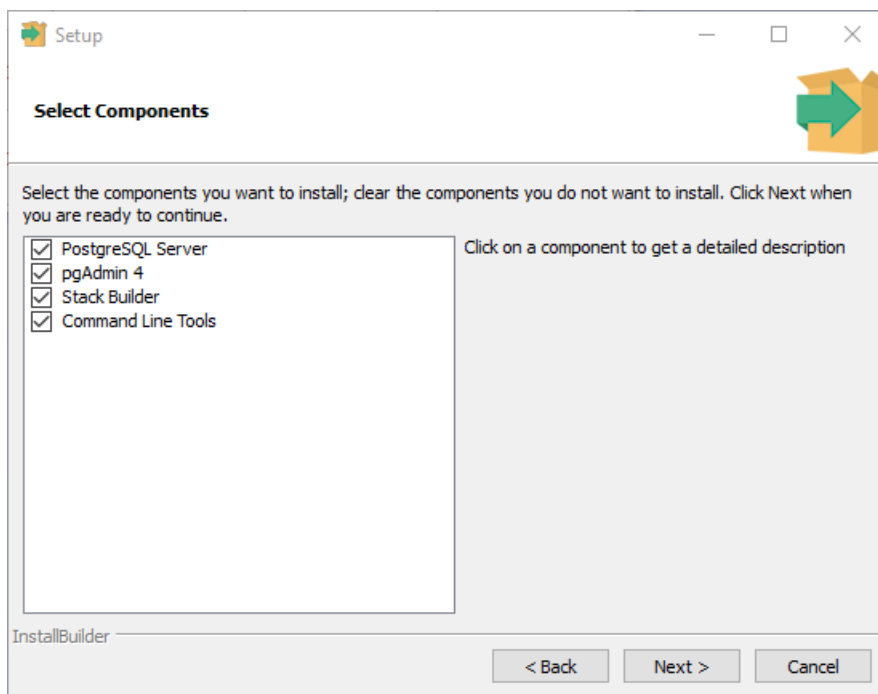
2. Execute the installer file. Click **next** to continue.



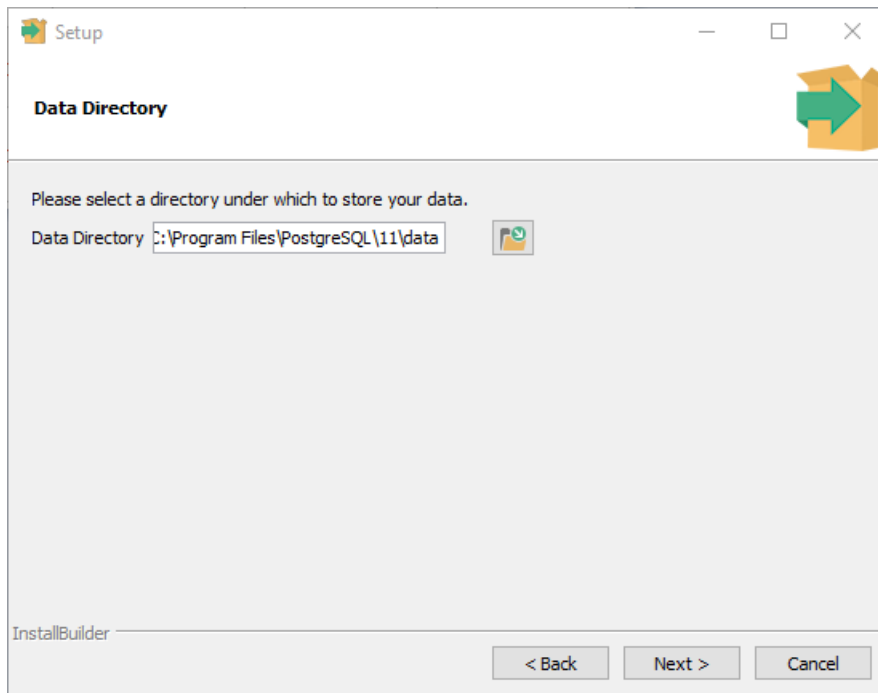
3. Configure the **installation directory**, *if necessary*. You may want to remember this installation directory. Click **Next >** to continue.



4. Choose the **components** you want, *it is recommended to install everything*. Click **Next >** to continue.

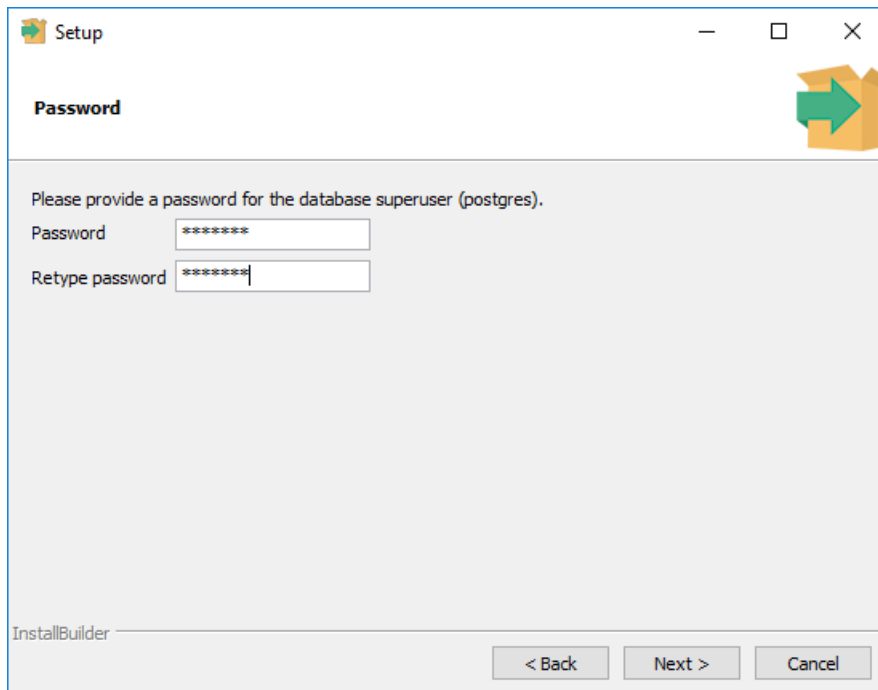


5. Configure the **data directory**, *if necessary*. Click **Next >** to continue.



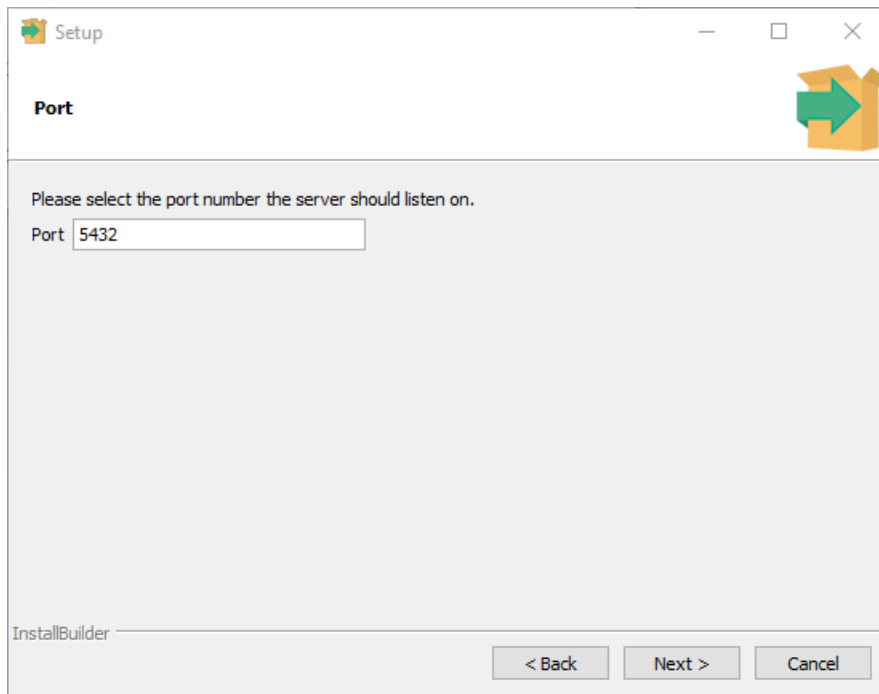
The screenshot shows a Windows-style setup window titled "Setup" with a green arrow icon in the top right corner. The window has a title bar with standard minimize, maximize, and close buttons. The main content area is titled "Data Directory" and contains the instruction "Please select a directory under which to store your data." Below this, there is a text box labeled "Data Directory" containing the path "C:\Program Files\PostgreSQL\11\data". To the right of the text box is a small icon representing a folder. At the bottom of the window, there is a status bar that says "InstallBuilder" and three buttons: "< Back", "Next >", and "Cancel".

6. Create a **password** for the default database superuser named "postgres". Retype the **password** and click **Next >** to continue.



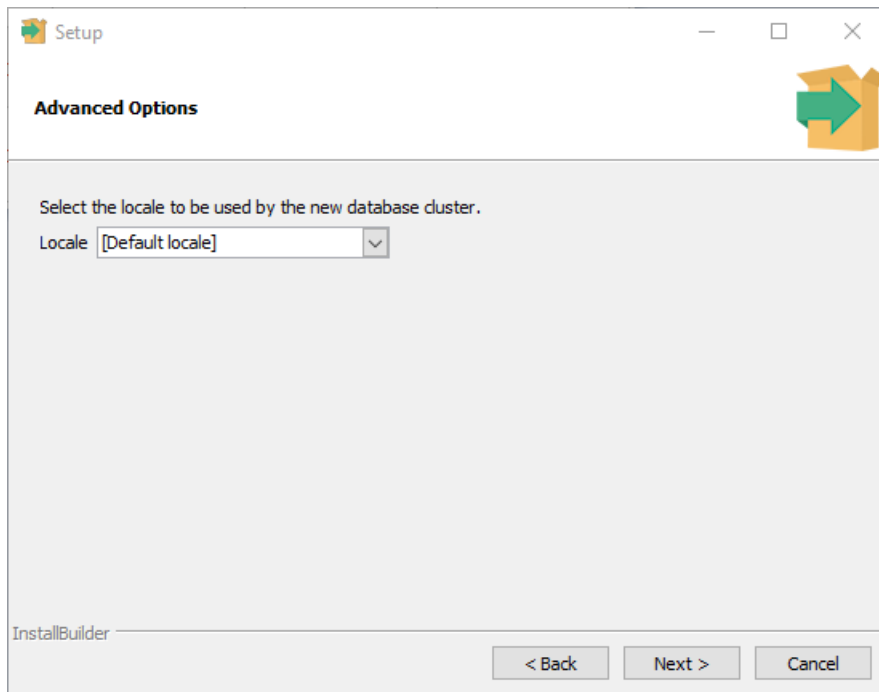
The screenshot shows a Windows-style setup window titled "Setup" with a green arrow icon in the top right corner. The window has a title bar with standard minimize, maximize, and close buttons. The main content area is titled "Password" and contains the instruction "Please provide a password for the database superuser (postgres)." Below this, there are two text boxes. The first is labeled "Password" and contains a series of asterisks. The second is labeled "Retype password" and also contains a series of asterisks. At the bottom of the window, there is a status bar that says "InstallBuilder" and three buttons: "< Back", "Next >", and "Cancel".

7. Configure the **port number**, *if necessary*. Click **Next >** to continue.



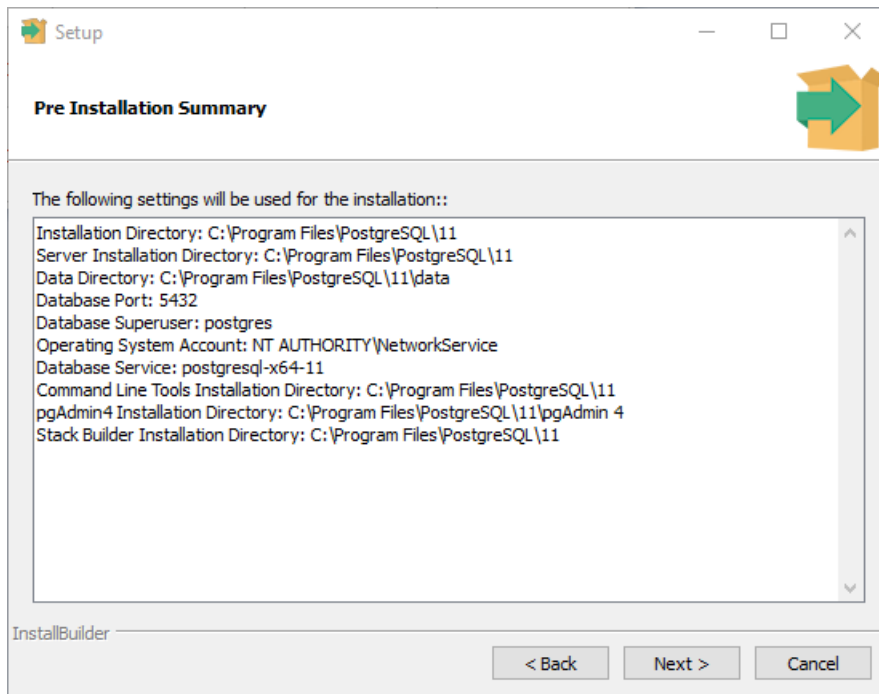
The screenshot shows a window titled "Setup" with a close button and a maximize button. In the top right corner, there is a green arrow pointing right inside a yellow box. The window has a section titled "Port". Below the title, it says "Please select the port number the server should listen on." There is a text input field labeled "Port" containing the value "5432". At the bottom of the window, there is a footer area with the text "InstallBuilder" on the left and three buttons: "< Back", "Next >", and "Cancel".

8. Configure the **locale**, *if necessary*. Click **next** to continue.

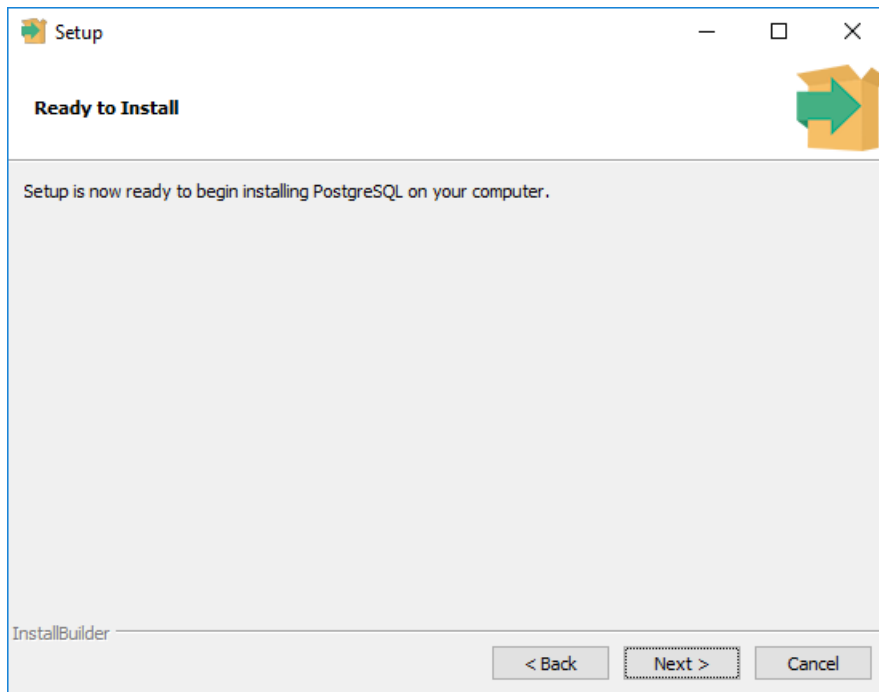


The screenshot shows a window titled "Setup" with a close button and a maximize button. In the top right corner, there is a green arrow pointing right inside a yellow box. The window has a section titled "Advanced Options". Below the title, it says "Select the locale to be used by the new database cluster." There is a dropdown menu labeled "Locale" with the value "[Default locale]". At the bottom of the window, there is a footer area with the text "InstallBuilder" on the left and three buttons: "< Back", "Next >", and "Cancel".

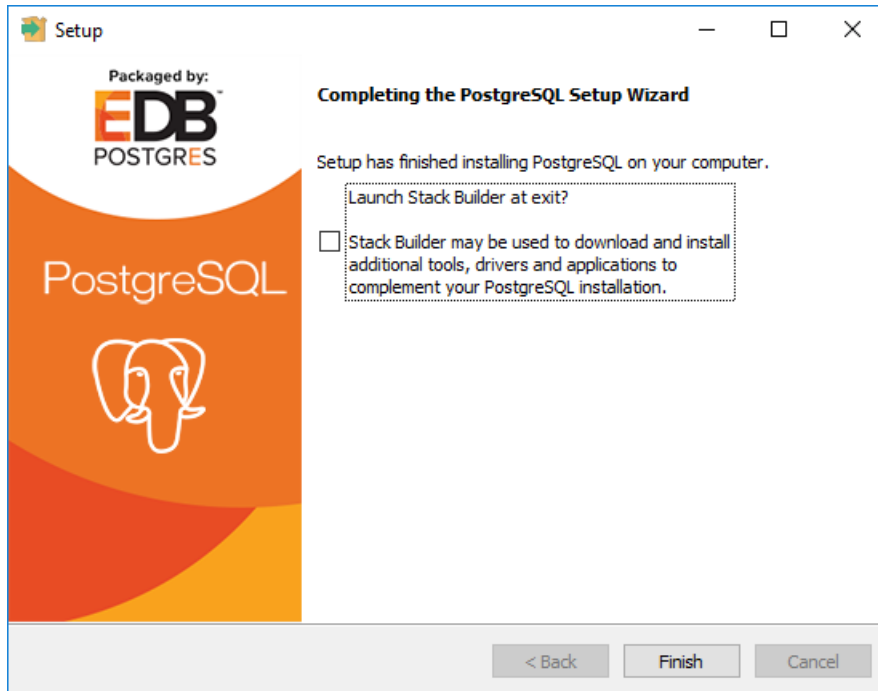
9. **Verify** that the configuration is OK. Click **Next >** to continue



10. Click **Next >** to install PostgreSQL on your computer.



On completion, both the PostgreSQL server as well as its command-line tool `psql` will be installed on your computer. **Untick** the box to launch Stack Builder at exit. Click **Finish** to complete.



### 1.1.1 Troubleshooting

If you encounter an error with the following message

**Problem running the post-install step. Installation may not complete correctly.**

Follow the steps below:

- 1 Uninstall PostgreSQL
- 2 Run command: `net user postgres /delete`
- 3 Click: **Control Panel -> User Accounts -> Configure advanced user profile properties**
  - a. Delete all "Unknown User" instances that seem to be left from PostgreSQL installation
- 4 Run: `compmgmt.msc`
  - a. Click: **Local Users and Groups -> Users -> New User...**
    - i. Name: postgres
    - ii. Password: <new password here>
    - iii. Click: Create
  - b. Right-click: **Local Users and Groups -> Users -> postgres**
    - i. Click: Member of
    - ii. Click: Add...

- iii. Type: Administrator
  - iv. Click: OK
- 5 Copy the PostgreSQL installer to C:\
- 6 Run: runas /user:postgres cmd.exe
- 7 Type: cd C:
- 8 Run: <postgresql\_installer>.exe
  - a. This should let you install PostgreSQL without any error
  - b. Check data folder and confirmed files are created successfully
- 9 Run: compmgmt.msc
  - a. Right-click: Local Users and Groups -> Users -> postgres
    - i. Click: Member of
    - ii. Click: Administrator
    - iii. Click: Remove
  - b. Right-click: Local Users and Groups -> Users -> postgres
    - i. Click: Member of
    - ii. Click: Add...
    - iii. Type: Power Users
    - iv. Click: OK

## 1.2. Linux

You can install PostgreSQL using your Linux distribution's package manager. User input will be highlighted in grey while prompts will be in monospace. For example, on Ubuntu Linux, the installation command is:

```
sudo apt update; sudo apt install postgresql postgresql-contrib
```

The installation creates a default database named "postgres" and a PostgreSQL user (role in PostgreSQL parlance) named "postgres". For convenience, create a new PostgreSQL user with the same name as your Linux login id; the following example assumes that your Linux userid is "alice".

```
$ sudo -u postgresql psql
[sudo] password for alice: *****
psql (10.6 (Ubuntu 10.6-0ubuntu0.18.14.1))
Type "help" for help.
```



```
postgres=# create role alice with superuser login;  
CREATE ROLE  
postgres=# \q
```

Now you can simply use the command `psql postgres` to access the database named “postgres” as the user “alice” without being prompted for your password<sup>1</sup>.

## 2. Connecting to PostgreSQL

There are two main ways to connect to a running PostgreSQL server. The first approach is to use PostgreSQL’s command-line tool named `psql` which is already installed along with the PostgreSQL server installation. Section 2.1 gives an introduction on how to use `psql`. The second approach is to use a graphical user interface. One such tool is named DBeaver<sup>2</sup>. We will not teach you how to use DBeaver but you are welcomed to use it. You may refer to the Section 3 for more information about DBeaver. In all our example in the module, we will use `psql` instead.

### 2.1. Using psql

`psql` provides an interactive terminal interface to edit/execute SQL commands/queries and view query results. On a Windows installation, you can find a short-cut to `psql` in the Start Window. Clicking this application will open a new command window with `psql` connected to the default database named “postgres”.

At `psql`’s prompt “postgres=#”, you can type in SQL commands. Input lines that terminate with a semicolon will be sent to the database server for execution. To exit `psql`, enter `\q` at `psql`’s “postgres=#” prompt. An example run is shown below.

```
postgres=# create table students (  
postgres=#     sid integer,  
postgres=#     name varchar(80)  
postgres=# );  
CREATE TABLE  
postgres=#
```

Besides SQL commands, you can also type in meta-commands that will be processed by `psql`. Each meta-command begins with an unquoted backslash. Any SQL command that has been typed but has not yet

---

<sup>1</sup> <https://www.postgresql.org/docs/current/auth-peer.html>

<sup>2</sup> <https://dbeaver.io/>

been sent for execution is stored in a memory buffer called *query buffer*. The contents of this buffer can be edited by invoking a configurable text editor within `psql`. The default editor is `vim` (on Linux<sup>3</sup>).

The table below shows some of the basic meta-commands.

Meta-command	Meaning
<code>\q</code>	Quit <code>psql</code> .
<code>\h</code>	Display all SQL commands with available syntax help.
<code>\h COMMAND</code>	Display syntax of <code>COMMAND</code> (e.g., <code>\h create table</code> ).
<code>\d</code>	List all created tables.
<code>\d TABLE</code>	List information on relation named <code>TABLE</code> .
<code>\p</code>	Display the contents of the query buffer; if the current query buffer is empty, display the most recently executed query.
<code>\w FILE</code>	Output the contents of the query buffer to the file named <code>FILE</code> ; if the current query buffer is empty, output the most recently executed query to <code>FILE</code> .
<code>\r</code>	Clear the query buffer.
<code>\e</code>	Invoke the text editor to edit the contents of the query buffer; if the current query buffer is empty, edit the most recently executed query.
<code>\e FILE</code>	Invoke the text editor to edit the contents of the file named <code>FILE</code> . The contents of the edited file will be copied to the query buffer at the end of the edit session.
<code>\o FILE</code>	Enable future query results to be saved to the file named <code>FILE</code> .
<code>\g</code>	Send the contents of the current query buffer to the server for execution; if the current query buffer is empty, the most recently sent query is re-executed.
<code>\i FILE</code>	Read the contents from the file named <code>FILE</code> and send their contents to the server for execution.
<code>\!</code>	Escapes from the <code>psql</code> session to a sub-shell. The <code>psql</code> session resumes when the sub-shell is exited.

The following example illustrates the execution of some meta-commands. Download the SQL script named `pizza.sql` from Luminus' SQL workbin.

```
postgres=# \d
          List of relations
 Schema |   Name   | Type  | Owner
-----+-----+-----+-----
 public | students | table | postgres
(1 row)
```

```
postgres=# \d students
          Table "public.students"
 Column |          Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
  sid   | integer                |           |          |
  name  | character varying(80)  |           |          |
```

---

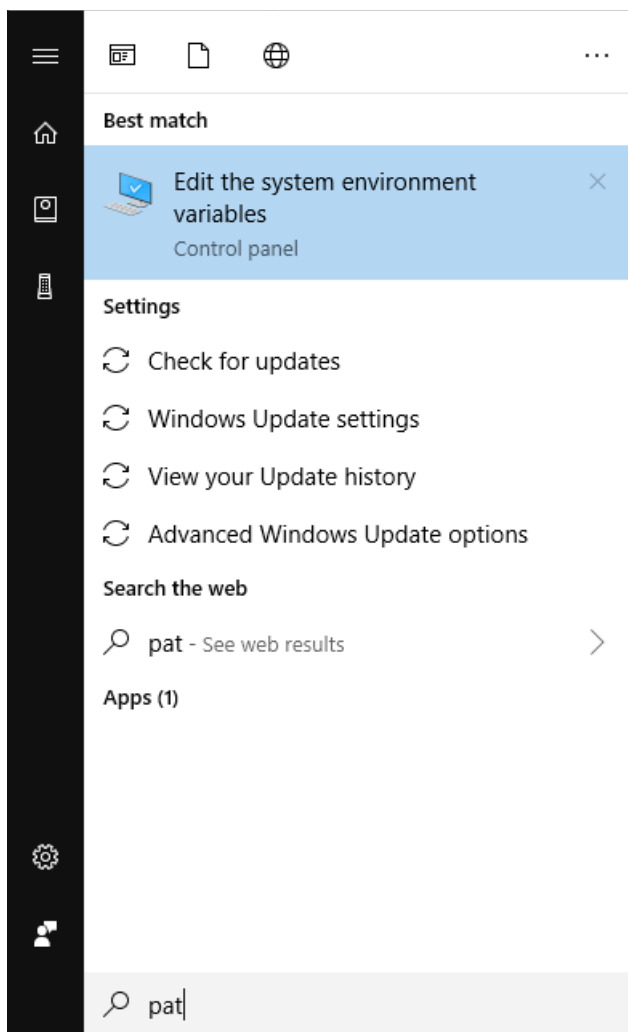
<sup>3</sup> The editor program can be configured using the environment variable `PGSQL_EDITOR`. For more details on configuring `psql`, refer to <https://www.postgresql.org/docs/current/app-psql.html>

```
postgres=# \i /PATHNAME-OF-DOWNLOADED-SCRIPT/pizza.sql
postgres=#
postgres=# \q
```

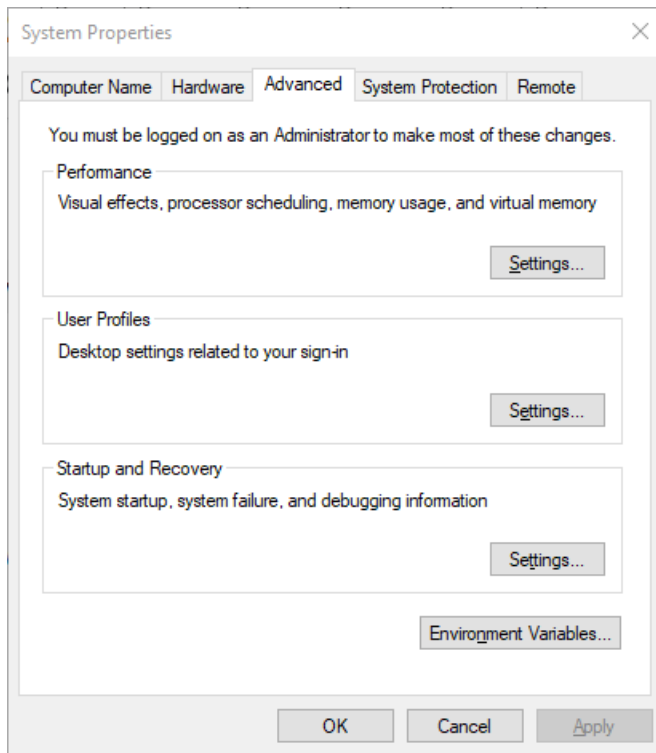
## 2.2. On Windows

It is likely that the above example does not work for Windows. If you are using Windows, it is advisable to add the bin directory of PostgreSQL into your PATH environment variable. To add bin directory of PostgreSQL to PATH environment variable, follow the steps below:

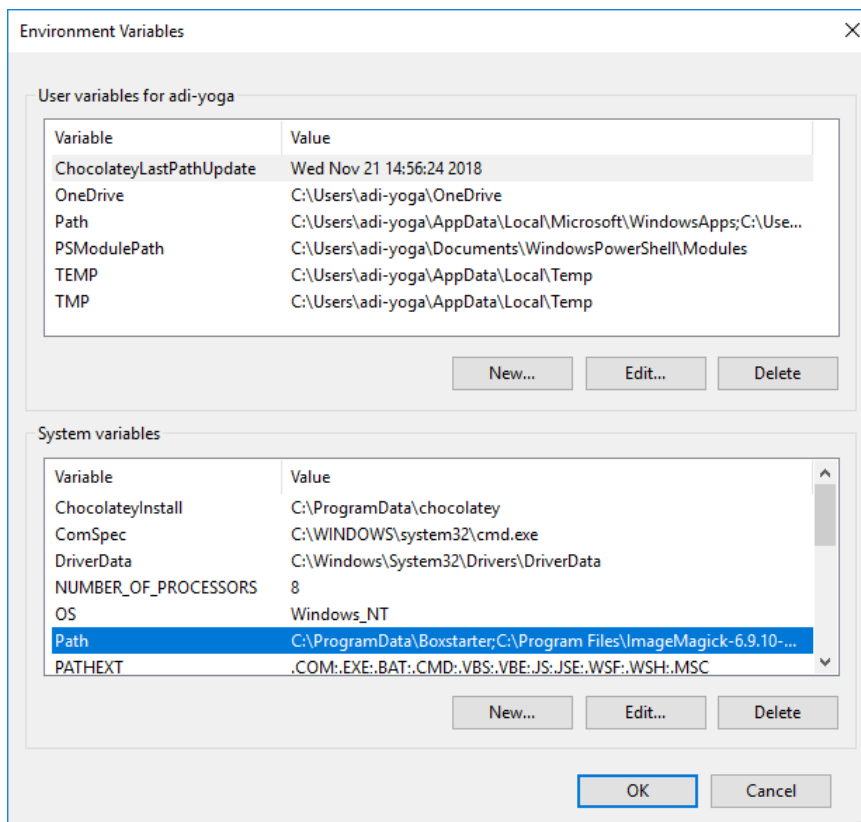
- Search for “**Edit the system environment variables**” in the Start Window.



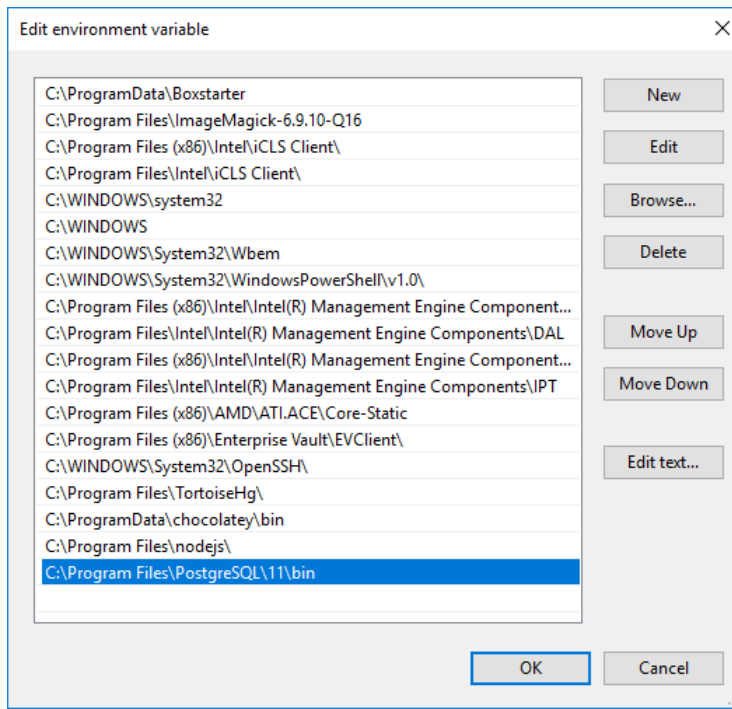
- Click on **Environment Variables...** button.



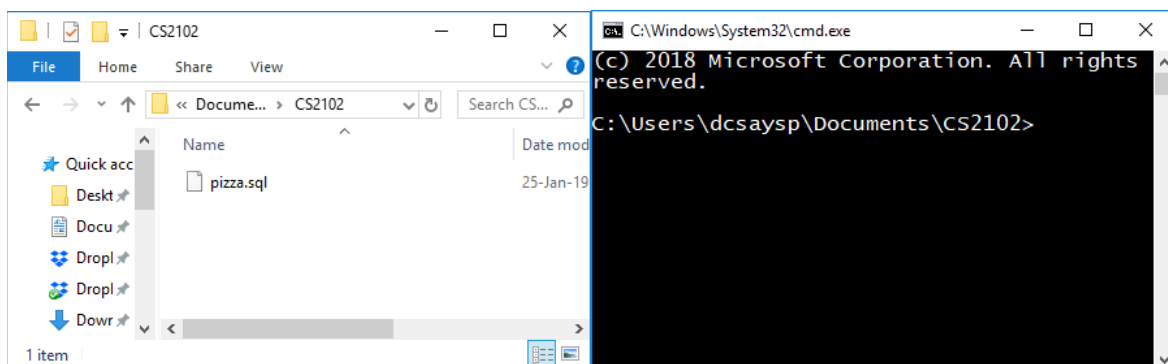
- Select **Path** on **System variables** panel and click the **Edit...** button.



- Click **New** and add the path to bin. If you use the default, it is likely to be C:\Program Files\PostgreSQL\11\bin. Otherwise, you should consult the step 3 on Section 1.1. Click **OK** to close.



Once you have added the bin directory of PostgreSQL to PATH environment variable, open CMD and go to the folder where you have downloaded pizza.sql.



Execute the following command on CMD.

```
psql -U username -d database
```

Where username is your username (or postgres for default) and database is your database (or postgres for default). You will be prompted for your password. Enter your password.

```
C:\Windows\System32\cmd.exe - psql -U postgres...
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\dcsaysp\Documents\CS2102>psql -U p
ostgres -d postgres
Password for user postgres:
```

You can now run the command `\i pizza.sql`.

```
C:\Windows\System32\cmd.exe - psql -U postgres...
reserved.

C:\Users\dcsaysp\Documents\CS2102>psql -U p
ostgres -d postgres
Password for user postgres:
psql (11.5)
WARNING: Console code page (437) differs fr
om Windows code page (1252)
        8-bit characters might not work co
rrectly. See psql reference
        page "Notes for Windows users" for
        details.
Type "help" for help.

postgres=# \i pizza.sql
```

### 3. References

- PostgreSQL documentation <https://www.postgresql.org/docs/current/index.html>
- PostgreSQL tutorial <https://www.postgresql.org/docs/current/tutorial.html>
- Manpage of psql <https://www.postgresql.org/docs/current/app-psql.html>
- DBeaver documentation <https://dbeaver.io/docs/>