Resources: Installing PostgreSQL

To install PostgreSQL on your computer, follow the below steps:  
  
Although I use *PSequel GUI* in my videos in this section, I recommend using <https://dbeaver.io/> which is open source and works on both Mac and Windows machines or <https://www.beekeeperstudio.io/>.  
  
**Mac:**Follow my previous video for instructions. You will need [homebrew](https://brew.sh/) for the easies way to set up. keep in mind you may have to run with the 'sudo' command.

[PSequel GUI](http://www.psequel.com/)  
  
**Windows:**Follow [this tutorial](http://www.postgresqltutorial.com/install-postgresql/)  
  
The most common issue you will find is this: <https://stackoverflow.com/questions/11460823/setting-windows-path-for-postgres-tools>. This means you will just need to set the path correctly for PostgreSQL:  
Step 1:) Goto Control Panel\System and Security\System

Step 2:) Click on Advanced system settings

Step 3:) Click on environment Variables

Step 4:) Select path and select edit

Step 5:) click browse and go to the install dir of PostgreSQL and make sure to go into the bin folder (Usually C:\Program Files\PostgreSQL\9.5\bin)

Step 6:) Click Ok on all open windows

Step 7:) restart  your command terminal  
  
  
Finally, for Windows users if you go to the **QA forum** of the previous video and this lecture, people have posted tips and tricks in case you are running into issues.

**Linux:  
  
See the attached resource (psql installation.docx)  Thanks to fellow student Saso Trpovski for the contribution!  
  
OR another option:***Thanks to fellow student Dimitris for this great guide:*

For any of the Linux users following the course and interested in installing PostgreSQL along with a GUI (eg. pgAdmin), [their website has wonderful instructions](https://www.postgresql.org/download/), and so does their wiki (for example, [this link is for Debian and Ubuntu based distros](https://wiki.postgresql.org/wiki/Apt)).

Also, one way to issue the commands you typed in the video to start, stop, restart PostgreSQL in Linux is:

1. sudo systemctl start postgresql # starts the server
2. sudo systemctl stop postgresql # stops it
3. sudo systemctl restart postgresql # restart it
4. sudo systemctl status postgresql # check the server's status

The "createdb test " command and the "psql 'test' " command are the same (at least for Debian/Ubuntu systems) from what I saw.

When it's first installed, PostgreSQL just has the 'postgres' user, and the way to initially enter PostgreSQL is by typing  sudo su - postgres , and then psql .  After Andrei creates the 'test' database, we can create a user with the same name as our current logged in user, to be a database administrator. This way we can just type in psql 'test'  from the command line and enter the database without the need of logging in as the 'postgres' user, just like Andrei does in the lecture. This can be done with CREATE USER your-user-name-here WITH SUPERUSER; , and we can verify that he was created with \du . Now we can exit by typing \q  and then exit , and enter our database just like Andrei does, with psql 'test' .

Lastly, with pgAdmin4 we need to create a connection with the server the first time we use it, and this is done by right-clicking 'Servers' on the left pane, and choosing 'Create' > 'Server'. We give our server a name, and in the 'Connection' tab we type in 'localhost' as the host, just like Andrei shows in the lecture, and press 'Save'.

**Troubleshooting: (Thank you to contribution from Samuel)**

Issue: Homebrew install of Postgresql will not execute successfully. $ brew link postgresql  results in failure due to directory not writable. New version of Homebrew will not allow sudo commands and System Integrity Protection prevents changing permissions.

Details: I tried to use homebrew to install postgres and kept running into issues with syslink. When I ran $ brew link postgresql  as homebrew suggested, I kept running into an error that it couldn't be completed because certain folders were not writable. I thought this would be easily remedied by running sudo but unfortunately the most current version of homebrew no longer allows the use of sudo commands due to security risks. My next thought was to my root user and use the macOS GUI interface to change the permissions on this folder because I am not sure how to do this on the terminal. Regardless of being logged in as 'root,' the OS would not let me change the permissions of the folder. I also attempted to use sudo and change the permissions in terminal and it did not work either. After several days of banging my head against the wall try all kinds of things  to find a solution, I discovered that since El Capitan, macOS introduced System Integrity Protection aka 'SIP' or 'rootless.' As it turned out, once I disabled SIP, logged back into 'root' and changed my regular accounts permissions to Read/Write on the problem directories, I was able to go back to my regular account and successfully execute $ brew install postgresql . Now it works and I'm so happy to be returning to my lessons!

*Steps to Resolve: (WARNING! these steps require you to mess with very critical stuff on your computer. it is best to exercise EXTREME caution when performing this and to revert the system back to its secure state at the end)*

Assuming you currently have postgresql installed through homebrew but unable to link due a scenario like the one mentioned above, here is what I suggest to resolve your issue...

1. Run $ brew link postgresql   
2. Write down the directory path that the error says it is not able to write to. (e.g. usr/local/share/man/man7) NOTE: you'll want to actually write this down on paper or take a picture of the screen on your phone because you will not be able to use copy and paste)  
3. Enable your 'root' user account if you have not already done so. (https://support.apple.com/en-us/HT204012) NOTE: make sure to make a really good password for this account and write it down somewhere safe. This is a powerful account and there's no way to recover the password.  
4. Disable System Integrity Protection. (https://www.igeeksblog.com/how-to-disable-system-integrity-protection-on-mac/)  
5. Log into 'root' user account  
6. In Finder menu bar select GO > GO TO FOLDER... (CMND + SHFT + G) and type in the path from Step 2.  
7. Right-Click/ Cntrl-Click the folder and select Get Info  
8. Click the plus sign at the bottom of Sharing & Permissions  
9. Add your regular account to the list and change the permission to Read & Write  
10. Go back to your regular account, run $ brew uninstall postgresql , then $ brew update  and $ brew doctor . If those are all set run $ brew install postgresql .  
11. You should be able to install without any problems now. However, if you run into a linking and permissions problem again, run $ brew link postgresql to figure out the problematic directory and repeat Steps 5 - 10 with whatever other directories are giving you trouble.  
12. If everything is up and running properly. It is probably best to at least enable SIP again (instruction in the article linked in Step 4).

(To check that everything is working. I recommend running $ brew services start postgresql  then $ createdb 'test' . In my case, it was when I originally tried to run createdb and got "command not found" that I realized something was wrong.)