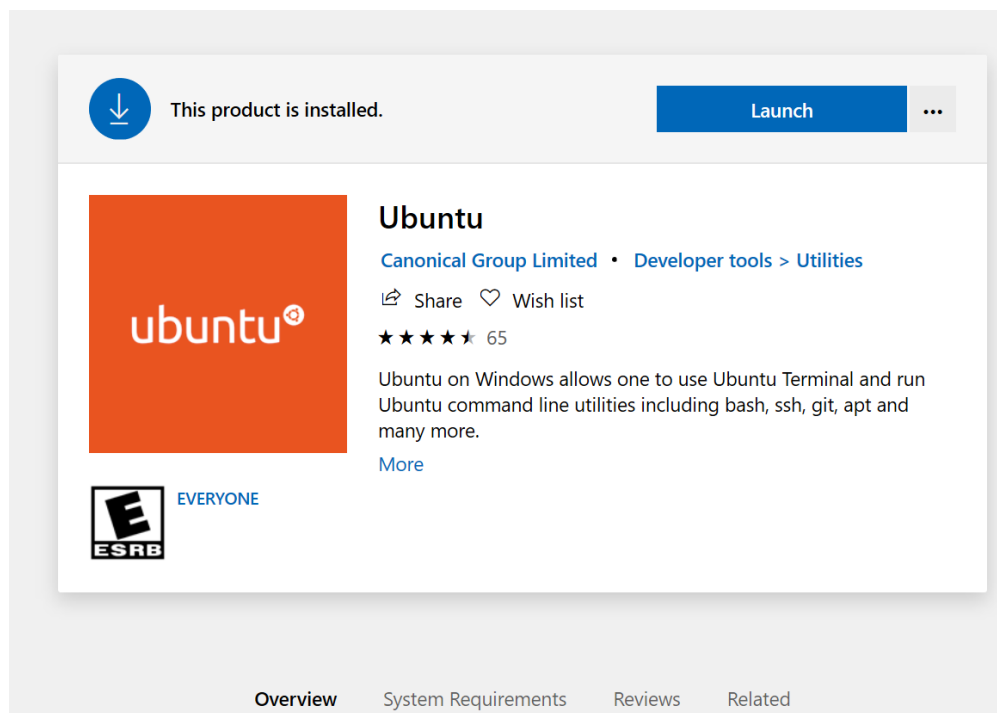


Note: If you are using Linux/Ubuntu on Virtual Machine then you can skip straight to Step 2.

Note: Do not copy paste commands mentioned below. Type each command out as there are some issues when you paste the command copied from this document directly.

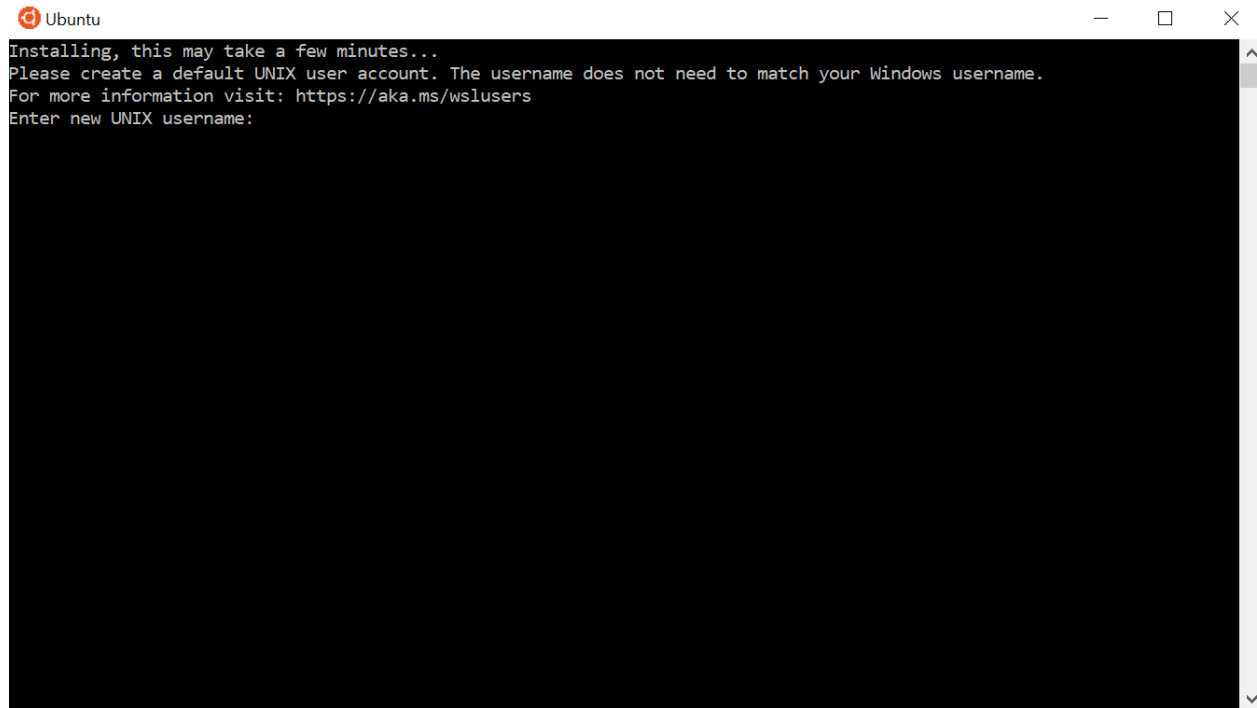
Step 1 (Only tested for Windows 10 users):

Install the Ubuntu App from Microsoft App store



Launch the application and you should see this screen:

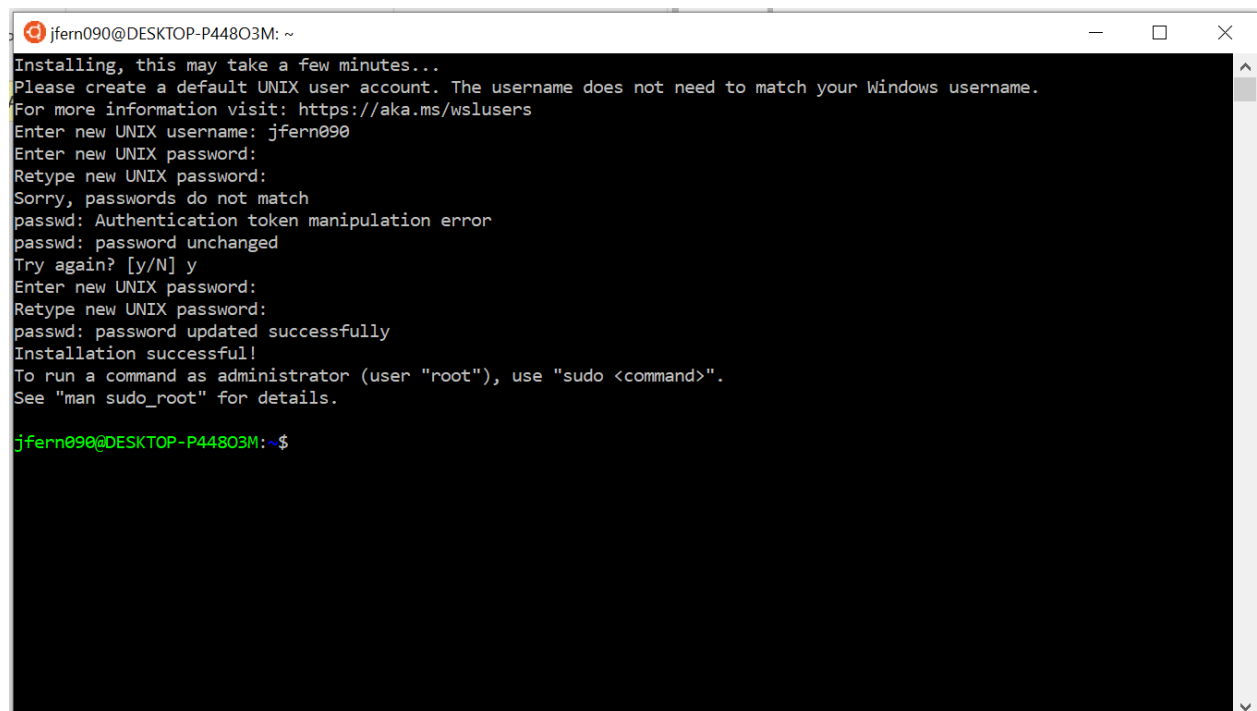
Note: Application can be launched from the Start up menu as well. It also takes a few mins to provide a prompt to request for the super user details.



A terminal window titled "Ubuntu" with standard window controls. The text inside the terminal is as follows:

```
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username:
```

Once you've entered your details you will be able to see a prompt with your username on it. In this case it is my uOttawa id:



A terminal window titled "jfern090@DESKTOP-P448O3M: ~" with standard window controls. The text inside the terminal is as follows:

```
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: jfern090
Enter new UNIX password:
Retype new UNIX password:
Sorry, passwords do not match
passwd: Authentication token manipulation error
passwd: password unchanged
Try again? [y/N] y
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

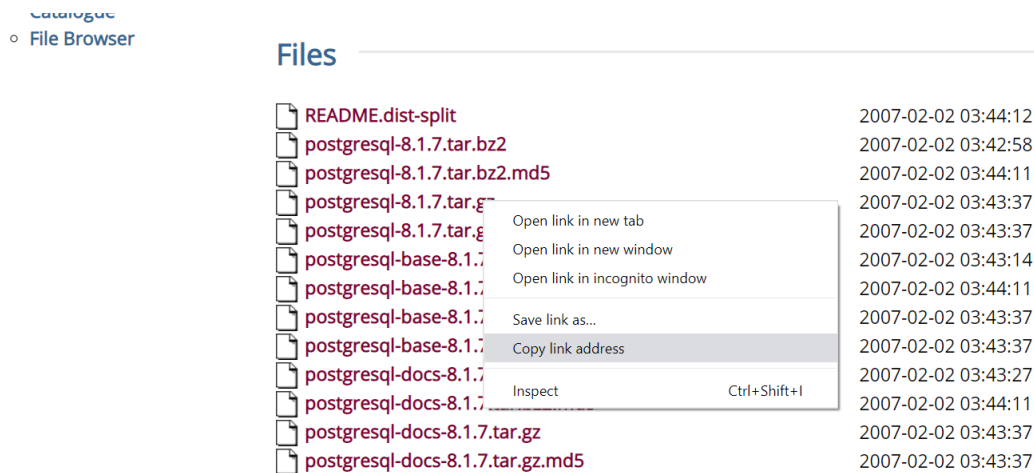
jfern090@DESKTOP-P448O3M:~$
```

Step 2. Download and extract postgresql source files (From here on these steps are for all users)

Simply go to the FTP server link : <https://www.postgresql.org/ftp/source/> on your favourite browser.

Scroll down to the version you would like to install. For this document I've used v8.1.7. You could also use v8.1.4.

Now right click on the postgresql-8.1.7.tar.gz file in the list and copy the link address:



Go back to the Ubuntu app and type the below command and link

`wget https://ftp.postgresql.org/pub/source/v8.1.7/postgresql-8.1.7.tar.gz`

```
jfern090@DESKTOP-P44803M: ~  
jfern090@DESKTOP-P44803M: ~$ wget https://ftp.postgresql.org/pub/source/v8.1.7/postgresql-8.1.7.tar.gz  
--2019-11-13 10:26:35-- https://ftp.postgresql.org/pub/source/v8.1.7/postgresql-8.1.7.tar.gz  
Resolving ftp.postgresql.org (ftp.postgresql.org)... 2620:122:b000:7::244, 2001:4800:3e1:1::246, 2a02:16a8:dc51::55, ...  
Connecting to ftp.postgresql.org (ftp.postgresql.org)|2620:122:b000:7::244|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 14627081 (14M) [application/x-gzip]  
Saving to: 'postgresql-8.1.7.tar.gz'  
  
postgresql-8.1.7.tar.gz      100%[=====] 13.95M  25.7MB/s  in 0.5s  
2019-11-13 10:26:36 (25.7 MB/s) - 'postgresql-8.1.7.tar.gz' saved [14627081/14627081]  
  
jfern090@DESKTOP-P44803M: ~$ ls  
postgresql-8.1.7.tar.gz  
jfern090@DESKTOP-P44803M: ~$
```

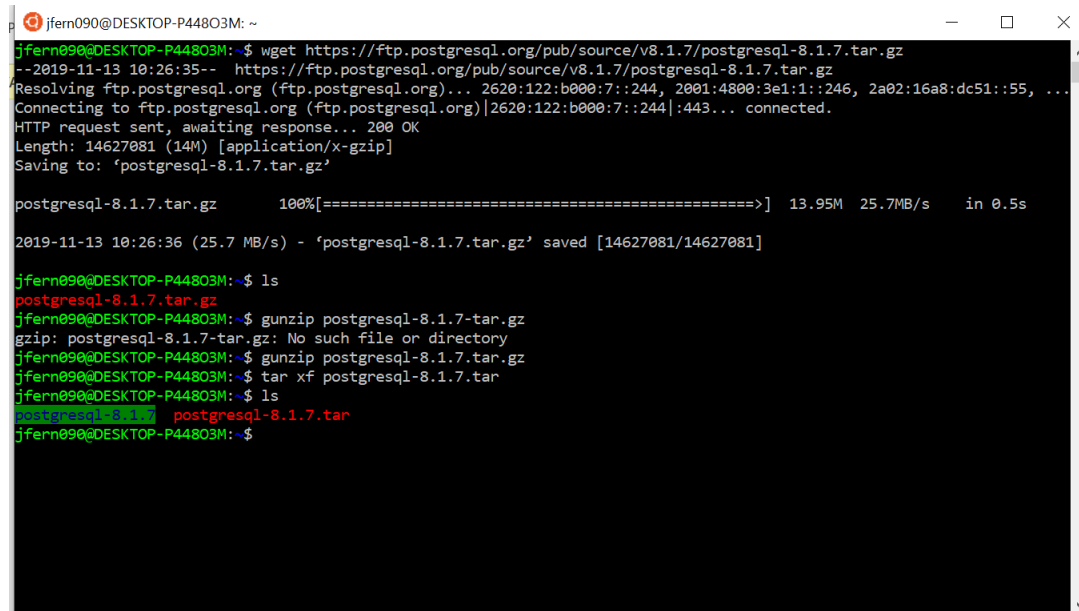
This step will download the file and you will be able to see it in your `/home/<user name>/` folder.

Next run these two commands to extract the source files:

```
gunzip postgresql-8.1.7.tar.gz
```

```
tar xf postgresql-8.1.7.tar
```

This will create a postgres-8.1.7 folder in the same destination:

A terminal window titled 'jfern090@DESKTOP-P44803M: ~' showing the process of downloading and extracting the postgresql-8.1.7 source files. The user runs 'wget https://ftp.postgresql.org/pub/source/v8.1.7/postgresql-8.1.7.tar.gz', which successfully downloads the file. Then, they run 'gunzip postgresql-8.1.7.tar.gz', which fails with the message 'No such file or directory'. Finally, they run 'tar xf postgresql-8.1.7.tar', which successfully extracts the files, creating a 'postgresql-8.1.7' directory. The terminal output is as follows:

```
jfern090@DESKTOP-P44803M:~$ wget https://ftp.postgresql.org/pub/source/v8.1.7/postgresql-8.1.7.tar.gz
--2019-11-13 10:26:35-- https://ftp.postgresql.org/pub/source/v8.1.7/postgresql-8.1.7.tar.gz
Resolving ftp.postgresql.org (ftp.postgresql.org)... 2620:122:b000:7::244, 2001:4800:3e1:1::246, 2a02:16a8:dc51::55, ...
Connecting to ftp.postgresql.org (ftp.postgresql.org)|2620:122:b000:7::244|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 14627081 (14M) [application/x-gzip]
Saving to: 'postgresql-8.1.7.tar.gz'

postgresql-8.1.7.tar.gz      100%[=====] 13.95M  25.7MB/s  in 0.5s

2019-11-13 10:26:36 (25.7 MB/s) - 'postgresql-8.1.7.tar.gz' saved [14627081/14627081]

jfern090@DESKTOP-P44803M:~$ ls
postgresql-8.1.7.tar.gz
jfern090@DESKTOP-P44803M:~$ gunzip postgresql-8.1.7.tar.gz
gzip: postgresql-8.1.7.tar.gz: No such file or directory
jfern090@DESKTOP-P44803M:~$ gunzip postgresql-8.1.7.tar.gz
jfern090@DESKTOP-P44803M:~$ tar xf postgresql-8.1.7.tar
jfern090@DESKTOP-P44803M:~$ ls
postgresql-8.1.7  postgresql-8.1.7.tar
jfern090@DESKTOP-P44803M:~$
```

Step 3: Install requisite libraries

Here's the list of commands for libraries that we found were not available in this Ubuntu application. If you are using Ubuntu/Linux on a VM you probably have these libraries already none the less, it's better to check before the next step.

```
sudo apt-get update
```

```
sudo apt-get install make
```

```
sudo apt-get install libreadline7 libreadline-dev
```

```
sudo apt-get install zlib1g zlib1g-dev
```

```
sudo apt-get install clang
```

Note: We have to use clang to compile this version of Postgres. An older version of GCC would work but the current GCC versions will throw some errors during compilation.

Step 4: Configure this version of postgresql

```
jfern090@DESKTOP-P44803M: ~/postgresql-8.1.7
jfern090@DESKTOP-P44803M:~$ ls
postgresql-8.1.7  postgresql-8.1.7.tar
jfern090@DESKTOP-P44803M:~$ cd postgresql-8.1.7
jfern090@DESKTOP-P44803M:~/postgresql-8.1.7$
```

Use this command to start configuration:

```
sudo ./configure CC="clang" CFLAGS="-O1" --enable-debug -enable-depend
```

if the above fails, try the option below

```
sudo ./configure CC=/usr/bin/clang CFLAGS="-O1"
```

***Note that it's a capital O and not a Zero**

This will throw an error in case we missed any other requisite libraries. In case everything goes well we will see this screen.

```
jfern090@DESKTOP-P44803M: ~/postgresql-8.1.7
checking for int8... no
checking for uint8... no
checking for int64... no
checking for uint64... no
checking for sig_atomic_t... yes
checking for POSIX signal interface... yes
checking for special C compiler options needed for large files... no
checking for _FILE_OFFSET_BITS value needed for large files... no
checking for _LARGE_FILES value needed for large files... no
checking for working memcmp... yes
checking for onsgmls... no
checking for nsgmls... no
checking for openjade... no
checking for jade... no
checking for DocBook V4.2... no
checking for DocBook stylesheets... no
checking for collateindex.pl... no
checking for sgmlspl... no
configure: creating ./config.status
config.status: creating GNUmakefile
config.status: creating src/Makefile.global
config.status: creating src/include/pg_config.h
config.status: linking ./src/backend/port/tas/dummy.s to src/backend/port/tas.s
config.status: linking ./src/backend/port/dynloader/linux.c to src/backend/port/dynloader.c
config.status: linking ./src/backend/port/sysv_sema.c to src/backend/port/pg_sema.c
config.status: linking ./src/backend/port/sysv_shmem.c to src/backend/port/pg_shmem.c
config.status: linking ./src/backend/port/dynloader/linux.h to src/include/dynloader.h
config.status: linking ./src/include/port/linux.h to src/include/pg_config_os.h
config.status: linking ./src/makefiles/Makefile.linux to src/Makefile.port
jfern090@DESKTOP-P44803M:~/postgresql-8.1.7$
```

Next, type this command:

```
sudo make
```

```
jfern090@DESKTOP-P448O3M: ~/postgresql-8.1.7
sort: option requires an argument -- 'o'
Try 'sort --help' for more information.
join: missing operand after '2.1'
Try 'join --help' for more information.
ranlib libplpgsql.a
clang -O1 -Wall -Wmissing-prototypes -Wpointer-arith -Winline -Wdeclaration-after-statement -Wendif-labels -fno-strict-aliasing -g -fpic -shared -Wl,-soname,libplpgsql.so.1 pl_gram.o pl_handler.o pl_comp.o pl_exec.o pl_funcs.o -L../src/port -o libplpgsql.so.1.0
rm -f libplpgsql.so.1
ln -s libplpgsql.so.1.0 libplpgsql.so.1
rm -f libplpgsql.so
ln -s libplpgsql.so.1.0 libplpgsql.so
make[4]: Leaving directory '/home/jfern090/postgresql-8.1.7/src/pl/plpgsql/src'
make[3]: Leaving directory '/home/jfern090/postgresql-8.1.7/src/pl/plpgsql'
make[2]: Leaving directory '/home/jfern090/postgresql-8.1.7/src/pl'
make -C makefiles all
make[2]: Entering directory '/home/jfern090/postgresql-8.1.7/src/makefiles'
make[2]: Nothing to be done for 'all'.
make[2]: Leaving directory '/home/jfern090/postgresql-8.1.7/src/makefiles'
make -C utils all
make[2]: Entering directory '/home/jfern090/postgresql-8.1.7/src/utils'
make[2]: Nothing to be done for 'all'.
make[2]: Leaving directory '/home/jfern090/postgresql-8.1.7/src/utils'
make[1]: Leaving directory '/home/jfern090/postgresql-8.1.7/src'
make -C config all
make[1]: Entering directory '/home/jfern090/postgresql-8.1.7/config'
make[1]: Nothing to be done for 'all'.
make[1]: Leaving directory '/home/jfern090/postgresql-8.1.7/config'
All of PostgreSQL successfully made. Ready to install.
jfern090@DESKTOP-P448O3M:~/postgresql-8.1.7$
```

Great! Now it's time to install it with this command

sudo make install

```
jfern090@DESKTOP-P448O3M: ~/postgresql-8.1.7
/bin/sh ../../config/install-sh -c pg_resetxlog /usr/local/pgsql/bin/pg_resetxlog
make[3]: Leaving directory '/home/jfern090/postgresql-8.1.7/src/bin/pg_resetxlog'
make[2]: Leaving directory '/home/jfern090/postgresql-8.1.7/src/bin'
make -C pl install
make[2]: Entering directory '/home/jfern090/postgresql-8.1.7/src/pl'
make[3]: Entering directory '/home/jfern090/postgresql-8.1.7/src/pl/plpgsql'
make -C src install
make[4]: Entering directory '/home/jfern090/postgresql-8.1.7/src/pl/plpgsql/src'
/bin/sh ../../config/install-sh -c -m 755 libplpgsql.so.1.0 /usr/local/pgsql/lib/plpgsql.so
make[4]: Leaving directory '/home/jfern090/postgresql-8.1.7/src/pl/plpgsql/src'
make[3]: Leaving directory '/home/jfern090/postgresql-8.1.7/src/pl/plpgsql'
make[2]: Leaving directory '/home/jfern090/postgresql-8.1.7/src/pl'
make -C makefiles install
make[2]: Entering directory '/home/jfern090/postgresql-8.1.7/src/makefiles'
mkdir -p -- /usr/local/pgsql/lib/pgxs/src/makefiles
/bin/sh ../../config/install-sh -c -m 644 ./pgxs.mk /usr/local/pgsql/lib/pgxs/src/makefiles/
make[2]: Leaving directory '/home/jfern090/postgresql-8.1.7/src/makefiles'
make -C utils install
make[2]: Entering directory '/home/jfern090/postgresql-8.1.7/src/utils'
make[2]: Nothing to be done for 'install'.
make[2]: Leaving directory '/home/jfern090/postgresql-8.1.7/src/utils'
make[1]: Leaving directory '/home/jfern090/postgresql-8.1.7/src'
make -C config install
make[1]: Entering directory '/home/jfern090/postgresql-8.1.7/config'
mkdir -p -- /usr/local/pgsql/lib/pgxs/config
/bin/sh ../config/install-sh -c -m 755 ./install-sh /usr/local/pgsql/lib/pgxs/config/install-sh
/bin/sh ../config/install-sh -c -m 755 ./mkinstalldirs /usr/local/pgsql/lib/pgxs/config/mkinstalldirs
make[1]: Leaving directory '/home/jfern090/postgresql-8.1.7/config'
PostgreSQL installation complete.
jfern090@DESKTOP-P448O3M:~/postgresql-8.1.7$
```

Great all done with this step.

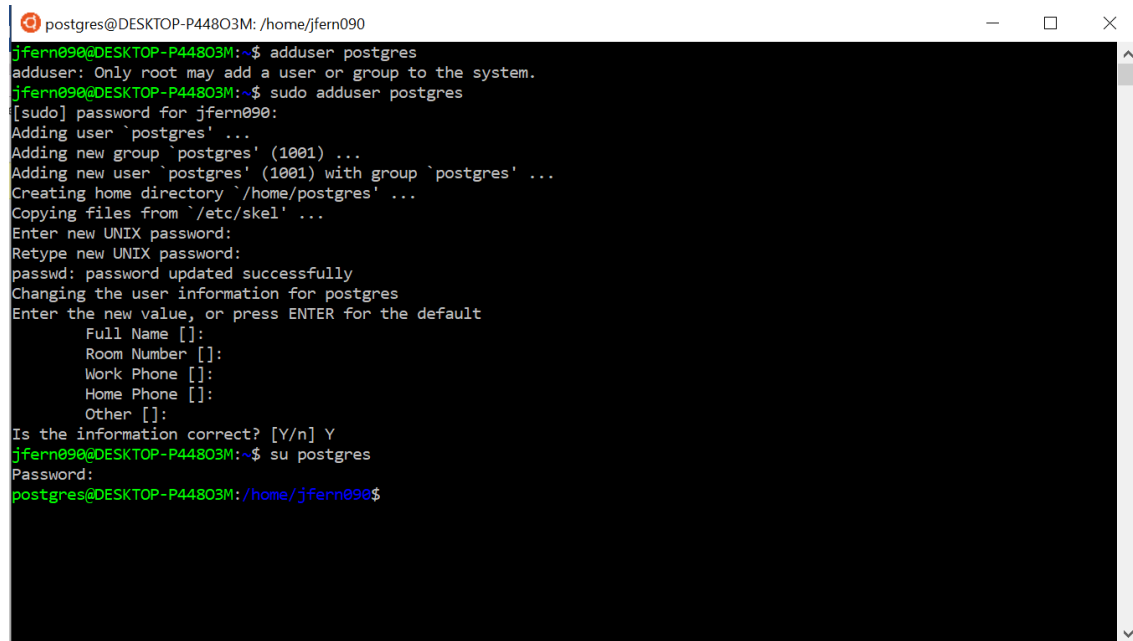
Step 5: Create postgres user and assign rights to files

Create a folder data in this path where all the files can be located:

```
sudo mkdir /usr/local/pgsql/data
```

```
sudo adduser postgres
```

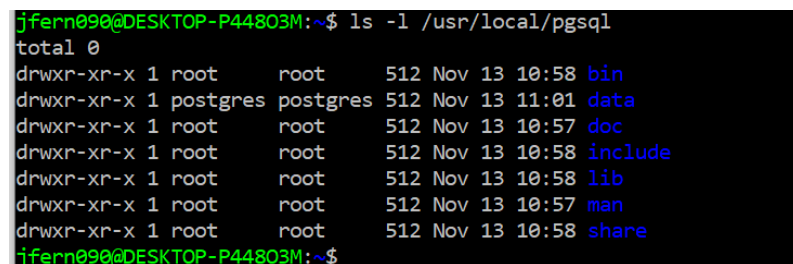
```
su postgres
```

A terminal window with a black background and green text. The prompt is 'postgres@DESKTOP-P448O3M: /home/jfern090'. The user enters 'adduser postgres'. The system responds with 'adduser: Only root may add a user or group to the system.' The user then enters 'sudo adduser postgres'. The system prompts for a password, which is entered. It then adds the user 'postgres' with group 'postgres' and a home directory at '/home/postgres'. It prompts for a new UNIX password, which is entered and confirmed. The system then prompts for additional information like full name, room number, etc., which are left blank. Finally, it asks 'Is the information correct? [Y/n] Y'. The user then enters 'su postgres' and the prompt changes to 'postgres@DESKTOP-P448O3M: /home/jfern090\$'.

By this step you will have created a new user. Now time to assign access to the new user for the data folder

```
Chown postgres /usr/local/pgsql/data/
```

Confirm that postgres user has access to the folder:

A terminal window with a black background and green text. The prompt is 'jfern090@DESKTOP-P448O3M:~\$'. The user enters 'ls -l /usr/local/pgsql'. The output shows a directory listing for '/usr/local/pgsql' with permissions, owner, group, size, date, and filename. The 'data' directory is owned by 'postgres' and has permissions 'drwxr-xr-x'. The prompt then changes to 'jfern090@DESKTOP-P448O3M:~\$'.

Step 6: Initialize DB and start the postgres server

Switch the user:

```
su postgres
```

Run this command to initialize the data folder with the necessary files

```
/usr/local/pgsq/bin/initdb -D /usr/local/pgsq/data/
```

```
postgres@DESKTOP-P448O3M: ~
creating directory /usr/local/pgsq/data/base ... ok
creating directory /usr/local/pgsq/data/base/1 ... ok
creating directory /usr/local/pgsq/data/pg_tblspc ... ok
selecting default max_connections ... 100
selecting default shared_buffers ... 1000
creating configuration files ... ok
creating template1 database in /usr/local/pgsq/data/base/1 ... ok
initializing pg_authid ... ok
enabling unlimited row size for system tables ... ok
initializing dependencies ... ok
creating system views ... ok
loading pg_description ... ok
creating conversions ... ok
setting privileges on built-in objects ... ok
creating information schema ... ok
vacuuming database template1 ... ok
copying template1 to template0 ... ok
copying template1 to postgres ... ok

WARNING: enabling "trust" authentication for local connections
You can change this by editing pg_hba.conf or using the -A option the
next time you run initdb.

Success. You can now start the database server using:

    /usr/local/pgsq/bin/postmaster -D /usr/local/pgsq/data
or
    /usr/local/pgsq/bin/pg_ctl -D /usr/local/pgsq/data -l logfile start

postgres@DESKTOP-P448O3M: ~$
```

Once you see the **Success** message you can then start the postgres server

Use this command to start the server:

```
/usr/local/pgsq/bin/pg_ctl start -D /usr/local/pgsq/data
```

```
postgres@DESKTOP-P448O3M:~$ /usr/local/pgsq/bin/pg_ctl start -D /usr/local/pgsq/data
postmaster starting
postgres@DESKTOP-P448O3M:~$ LOG: database system was shut down at 2019-11-13 12:20:52 STD
LOG: checkpoint record is at 0/38FF90
LOG: redo record is at 0/38FF90; undo record is at 0/0; shutdown TRUE
LOG: next transaction ID: 565; next OID: 10794
LOG: next MultiXactId: 1; next MultiXactOffset: 0
LOG: database system is ready
LOG: transaction ID wrap limit is 2147484146, limited by database "postgres"
```

Great! Now open another terminal / Ubuntu instance and run this command to first create a database and then access that database on the currently running postgres server.

```
/usr/local/pgsq/bin/create test
```

```
/usr/local/pgsq/bin/psql test
```



```
Select postgres@DESKTOP-P44803M: ~
jfern090@DESKTOP-P44803M:~$ su postgres
Password:
postgres@DESKTOP-P44803M:/home/jfern090$ cd ~
postgres@DESKTOP-P44803M:~$ /usr/local/pgsql/bin/createdb test
CREATE DATABASE
postgres@DESKTOP-P44803M:~$ /usr/local/pgsql/bin/psql test
Welcome to psql 8.1.7, the PostgreSQL interactive terminal.

Type: \copyright for distribution terms
      \h for help with SQL commands
      \? for help with psql commands
      \g or terminate with semicolon to execute query
      \q to quit

test=# \q
```

Great! Now you are able to access the server. Stop the server with this command

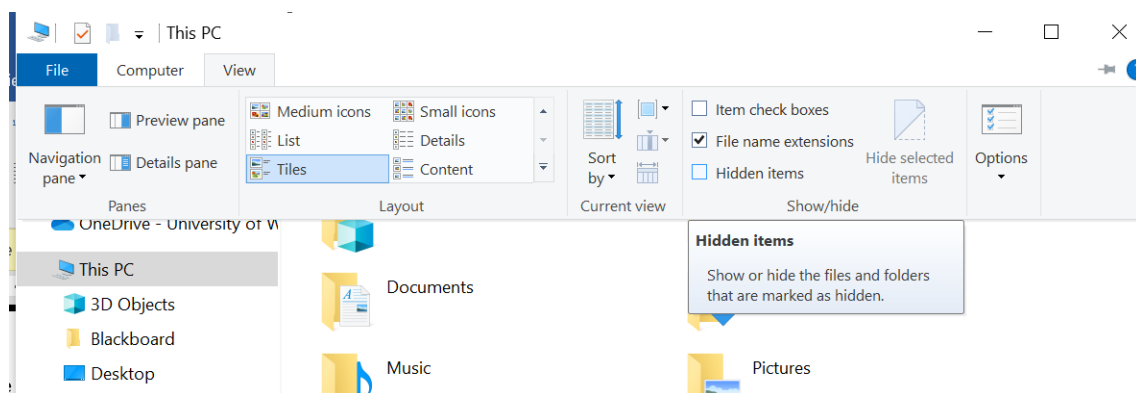
```
/usr/local/pgsql/bin/pg_ctl stop -mimmediate -D /usr/local/pgsql/data
```

Step 7: Add files from Windows Explorer (Only for users who are using this approach for the project. VM users can normally add their files normally)

I have added a schema.sql file to the Project folder. Use that to initialize the test database with the new tables. When adding files I've noticed Ubuntu terminal needs to be restarted so please do so before you do this step and start it again after you've added the file.

Here's how you can view your Ubuntu files from Windows Explorer.

First, display all Hidden files. Select/Check the Hidden items option here



Follow this link on the explorer:

```
C:\Users\<Windows user>\AppData\Local\Packages\CanonicalGroupLimited.UbuntuonWindows_79rhkp1fndgsc\LocalState\rootfs\home\postgres
```

You will notice within the rootfs file you can see all the related Ubuntu files. Here you can edit your C Files here. (which will be done a little later in this document).

Add the schema file here to this location. Start the Ubuntu terminal and postgres server and run this command to first reset the access rights to the schema file and then the next command to create all the table and rows in the test data base.

```
chmod 660 schema.sql
```

```
/usr/local/pgsql/bin/psql test -f schema.sql
```

```
postgres@DESKTOP-P448O3M: ~
postgres@DESKTOP-P448O3M:~$ chmod 660 schmea.sql
chmod: cannot access 'schmea.sql': No such file or directory
postgres@DESKTOP-P448O3M:~$ ls
schema.sql
postgres@DESKTOP-P448O3M:~$ clear
postgres@DESKTOP-P448O3M:~$ ls
schema.sql
postgres@DESKTOP-P448O3M:~$ chmod 660 schema.sql
postgres@DESKTOP-P448O3M:~$ ls -l
total 4
-rw-rw---- 1 postgres postgres 1526 Nov  7 17:23 schema.sql
postgres@DESKTOP-P448O3M:~$ /usr/local/pgsql/bin/psql test -f schema.sql
psql:schema.sql:4: NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "dept_pkey" for table "dept"
CREATE TABLE
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
psql:schema.sql:17: NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "emp_pkey" for table "emp"
CREATE TABLE
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
```

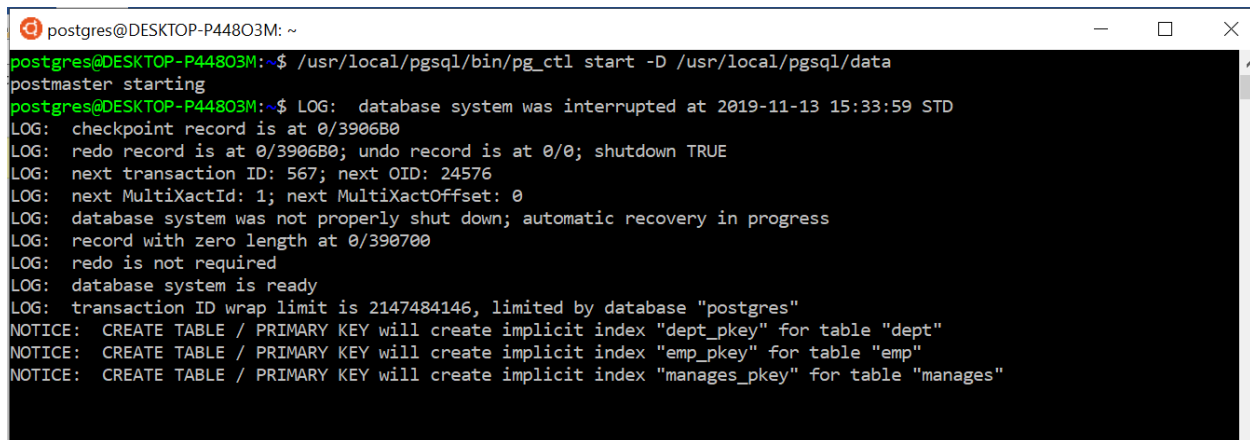
Test the existence of each table. Here's a snapshot for that.

```
postgres@DESKTOP-P448O3M: ~
postgres@DESKTOP-P448O3M:~$ clear
postgres@DESKTOP-P448O3M:~$ /usr/local/pgsql/bin/psql test
Welcome to psql 8.1.7, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help with psql commands
       \g or terminate with semicolon to execute query
       \q to quit

test=# select * from emp;
 eno |  ename  | dno
-----+-----+----
 101 | Smith   |    1
 201 | Kevin   |    2
 105 | Sally   |    1
 102 | Matt    |    1
 402 | Jeff    |    4
 205 | Amy     |    2
 401 | Tom     |    4
 202 | Alex    |    2
 103 | Sam     |    1
 302 | Joe     |    3
 304 | Sean    |    3
 206 | Martin  |    2
 203 | Simon   |    2
 104 | Jane    |    1
 501 | Max     |    5
 303 | Mike    |    3
 106 | Sarah   |    1
```

You will notice on the other terminal where you started your server will have these updates as well.

A terminal window titled 'postgres@DESKTOP-P448O3M: ~' with standard window controls. The terminal shows the command `/usr/local/pgsql/bin/pg_ctl start -D /usr/local/pgsql/data` and the output of the PostgreSQL server startup. The logs indicate a previous interruption and show the server starting successfully, including checkpoint records, transaction IDs, and notices about implicit index creation for tables 'dept', 'emp', and 'manages'.

```
postgres@DESKTOP-P448O3M: ~  
postgres@DESKTOP-P448O3M:~$ /usr/local/pgsql/bin/pg_ctl start -D /usr/local/pgsql/data  
postmaster starting  
postgres@DESKTOP-P448O3M:~$ LOG:  database system was interrupted at 2019-11-13 15:33:59 STD  
LOG:  checkpoint record is at 0/3906B0  
LOG:  redo record is at 0/3906B0; undo record is at 0/0; shutdown TRUE  
LOG:  next transaction ID: 567; next OID: 24576  
LOG:  next MultiXactId: 1; next MultiXactOffset: 0  
LOG:  database system was not properly shut down; automatic recovery in progress  
LOG:  record with zero length at 0/390700  
LOG:  redo is not required  
LOG:  database system is ready  
LOG:  transaction ID wrap limit is 2147484146, limited by database "postgres"  
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "dept_pkey" for table "dept"  
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "emp_pkey" for table "emp"  
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "manages_pkey" for table "manages"
```

Step 8: Change some C code and see the output.

This is the final step where we will update the postgres C code to just print out at what time it executed the symmetric hash join algorithm. This step assumes you have done some work on `nodeHash.c` and `nodeHashjoin.c` files in order to implement the symmetric hash join algorithm.

1. Change `postgresql.conf` settings by turning off these values. The `postgres.conf` file can be found in the `/usr/local/pgsql/data` folder

enable_mergejoin = off
enable_nestloop = off
2. On editing `postgresql.conf` the access to the file will change so make sure to run this command
`chmod 660 /usr/local/pgsql/data/postgresql.conf`
3. After you've added your let's say print statement in one of these file it's time to clean the `postgresql` files and re compile the codes. Run the *`sudo make clean`* command in the `postgresql-8.1.7` folder.

