CS452

PostgreSQL Install Instructions & Configuration

NOTE - These instructions are for installation on a Windows OS. Feedback from students indicate that installing on Linux and MAC OS’s is similar.

There are three steps to this assignment –

* Installing PostgreSQL
* Configuring PostgreSQL
* Loading the university database data

# 1.0: Installing PostgreSQL

PostgreSQL 9.4 (the latest stable version) should be compatible with all recent versions of Windows, Linux, and MAC OS platforms.

* If you have a doubt or if you try it out and something doesn’t work, you can check compatibility here.   
  <http://www.postgresql.org/docs/9.4/static/supported-platforms.html>  
  Start with version 9.4 and if your OS isn’t compatible, move down a version at a time (“This page in other versions:” in the upper left of the page).

After determining the PostgreSLQ version for your operating system, download and run the PostgeSQL installer from the EDB download website from

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

You can take the default options as you step through the wizard. You will be asked to supply a password for the database superuser (administrator) which is automatically created as part of the installation. The username of the superuser is “postgres”. Enter a password for the “postgres” superuser when prompted.

At the end of the install, you might be asked if you want to launch an application named “Stack Builder”. Uncheck the box to prevent it from launching. You won’t need it. Click “Finish” and you’re done.

All in all, the download and install should only take 5 minutes or so to download and install.

The PostgreSQL RDBMS runs as a service (daemon) and you will communicate with it through your localhost network connection at localhost:5432 (assuming you took the default options during the installation)

To verify that the service is running, do the following :

* Windows - Use the Administrative Tools->Services view to find the PostgreSQL service and verify its status is “Running”
* Linux (at least on Ubuntu) – in a terminal window run $ /etc/init.d/postgresql status  
  It should say the service is “on line”. If this doesn’t work on your Linux system, try the MAC OS method below.
* MAC OS X – in a terminal window run $ ps aux | grep postgres  
  If PostgreSQL is running you will see several “postgres” processes running (checkpointer, writer, etc)

You are now ready to use the administration tool to configure the database.

# 2.0: Configuration

In the configuration step we are going to do four things:

1. Connect to PostgreSQL using the pgAdmin III administration client
2. Create a normal Login Role to use with our projects
3. Create a database container for a new Login Role
4. Connect with the new Login Role and create the schema container for the university database data

## 2.1: Connect With pgAdmin III Administration Client

1. Locate and open the PostgreSQL administration client pgAdmin III.
2. In the Object Browser pane, expand the Servers tree node (if not already expanded).
3. You should see a server icon marked with a red ‘X’ indicating that you haven’t connected to the server yet. The label for the server node should be something like “PostreSQL 9.4 (localhost:5432)”. Right-click on the server icon to bring up the context menu and select Properties.
4. Fill out the properties page entries as follows -
   1. The Name, Host, and Port number entries should already be filled out. If not, enter the information –
      1. Name: PostgreSQL 9.4
      2. Host: localhost
      3. Port: 5432
   2. Leave the Service property blank.
   3. Enter “postgres” (without the quotes) for the name of the Maintenance DB if it’s not already filled in.
   4. Enter “postgres” again for the Username if it’s not already filled in.
   5. No other properties or tabs need to filled out right now. Click OK to close the properties dialog.
5. Right-click again on the server icon to bring up the context menu and this time select “Connect”
6. When challenged for a password, enter the password you created for the superuser during the installation process. Check the “Store password” checkbox to prevent being challenged each time you connect.
7. The red ‘X’ should disappear and the service node should expand to 4 sub-nodes, Databases(1), Tablespaces(2), Group Rules(0), and Login Roles(1). You are now connected to PostgreSQL as the superuser “postgres”.

## 2.3: Create a “Normal” User Login Role User – optional but recommended

It is possible to do all your work as the superuser, but it is not a good practice. The superuser account should be reserved for administrative tasks. It is highly recommended that you create another, non-superuser account to work in. If you decide not to do this, skip to ???.

In postgreSQL, user accounts are called Login Roles. We will create a non-superuser Login Role to use when doing normal database work and use the superuser account for administrative tasks.

We will do this in 5 steps -

1. While still connected to the server as the superuser, expand the Login Roles node. You should see “postgres” as the only Login Role listed. This is the account you are currently logged in under. Right-click on the Login Roles node to bring up the context menu and select New Login Role …
2. In the pop-up dialog, enter a name for the Role name. Your own username is a good suggestion.
3. Select the Definition tab and enter a password (twice) that you would like associated with the new Login Role. (It doesn’t have to be different than the superuser password)
4. No other values need to be entered in this dialog. Click OK to close. You should see a new Login Role under the Login Roles node. The rest of these instructions will refer to this new Login Role as the user role or user account.

### 2.4: Create a Database Container for the new Login Role

To keep work isolated from other accounts (if there were any), we will create a new database container that will belong to the new user account.

1. While still connected as the superuser, expand the Databases node. You should see a single database named “postgres” which belongs to the superuser by the same name.
2. Right-click on the Databases node to bring up the context menu and select New Database …
3. In the pop-up dialog, enter the name for a new database container. “cs452” works well, but it can be anything you choose, including the name of the user account, which is commonly done.
4. In the Owner drop-down list, select the Login Role you just created in the previous step.
5. No other information needs to be entered. Click OK to close the dialog and create the database container. You should see a new database, marked with a red ‘X’ and the name you just entered appear under the Databases node.

### 2.5: Connect as the User Account and Create a Schema Container for the University Database

We are still currently connected to the database as the superuser. Before creating the schema container, we must disconnect from the server and reconnect as the new user account.

#### 2.5.1: Connecting as the New Login Role

1. Right-click on the server node to bring up the context menu and select “Disconnect server”. The node collapses and displays a red ‘X’ to indicate that we are disconnected.
2. Right-click on the server node to bring up the context menu and select “Properties”. The properties dialog allows you to change the user account that will connect to the server.
3. Change the Username to be the name of the new user account name you just created. Leave the Maintenance DB information the same, i.e. “postgres”
4. Uncheck the Store password check box if it is checked.
5. Now click on the Advanced tab of the properties dialog.
6. To unclutter our view and prevent accidents, we are going to hide other databases (e.g. the superuser postgres database) and restrict this user account to just the database container we created for it. In the multi-line text box labeled “DB restriction”, enter the name of the new database container you created for the user account, surrounded by single quotes. For instance, if the name of the database container you just created is cs452, you would enter ‘cs452’ (including the single quotes).
7. Click OK to close the Properties dialog.
8. Now right-click on the server node to bring up the context menu and select Connect.
9. A pop-up dialog will appear prompting you for the password of the new Login Role. Enter the password and check the “Store password” checkbox to prevent future password challenges with connecting to the server. Click OK to connect to the server as user account.

#### 2.5.1: Create the University Schema

We will be using the database schema used in the textbook which models the (simplified) data needs of a fictitious university. The university database schema will be created within a schema container inside the user database container. To create the schema container, follow these steps –

1. While still connected as the user account, expand the Databases node. You should see just one database container, the one you created for the user account (e.g. cs452). The other database container is hidden because we restricted the user account to only be able to access its own database container. It will be marked with a red ‘X’, but that will disappear when you expand the node.
2. Expand the new database container node to reveal 5 subnodes.
3. Expand the Schemas node. You should see a single schema named “public”.
4. Right-click on the Schemas node to bring up the context menu and select “New Schema …”
5. In the Name text box, enter the name for the new schema container. I suggest using the name “university” (without the quotes).
6. In the Owner drop-down list, select the user name as owner.
7. Click OK to close the dialog and create the new schema.

We will now set this new schema container to be the default place to create and load tables.

In PostgreSQL, database Object ID’s (OID’s) are referenced using fully qualified names similar to package names in Java. The files we will use to create the university database schema and populate it with data do not have database/schema qualifiers on the OID’s. At this point, if you were to begin creating tables, they would default into the “public” schema instead of the “university” schema we just created. PostgreSQL maintains a schema search path that it uses when unqualified table names are used. We will change the default schema search path to default to the schema we just created. To change the default schema to be the “university” schema we want to use –

1. Click on the Query Tool icon in the tool ribbon – it looks like a magnifying glass with the letters “SQL” in the middle. The Query Tool window will appear.
2. In the SQL Editor pane, enter the following text, replacing role name and schema name with the user name and name of the schema we just created -  
     
   alter role «role name» set search\_path = ‘«schema name»’

For instance, if the user name I created is ***johnsonrc*** and the name of the new schema I just created is ***university***, I would enter

alter role johnsonrc set search\_path = ‘university’

1. Now click the plain green arrow (Execute query) icon in the tool bar ribbon. You should see a positive result message in the Messages tab of the Output pane.
2. Close the Query tool window – it will prompt you to save changes … just click “No”. It’s not asking you to save the changes you made with your “alter role” command; it’s asking if you want to save the text of the command so you could run it again later.

Your PostgreSQL installation is now configured and ready to load the example university database.

# 3.0: Loading the University Database

The commands to create the schema tables and insert the university data are stored in files in the LearningSuite : Content->Database Files->University Database folder. There are three files

* DDL.sql – use this file to create the schema tables in a newly created schema container
* DDL+drop.sql – use this file to drop all the tables and recreate them – essentially “reset” the schema tables
* smallRelationsInsertFile.sql – use this file to insert the small university data set

Download and store these files somewhere.

## 3.1: Create the Schema Tables

1. In pgAdmin III, open up the Query Tool window (the magnifying glass on the tool bar ribbon)
2. Select File->Open … from the main menu
3. In the “Open query file” dialog, select the DDL.sql file you just downloaded and click “OK”.
4. This will load the contents of the file into the SQL Editor pane of the Query Tool. You can take a look at the create table commands in the file. You will notice some constraints we haven’t talked about (e.g. ‘check (year > 1701 and year < 2100’)).
5. Execute the commands in the file by clicking the Execute query (plain green arrow) icon in the tool bar ribbon. You should see a reassuring message in the Output pane of the Query Tool window.
6. Without closing the Query Tool window, go back to pgAdmin III and expand the university schema container. You should see a bunch of subnodes. Verify that there are now 11 tables under the Tables node.

## 3.2: Insert the University Database Data

1. Go back to the Query Tool window and open the smallRelationsInsertFile.sql file into the Query editor.
2. Take a quick look at the file contents. You will notice that first all the rows are deleted from all the tables. Then there are lots of insert statements to insert the data back into the tables. If you haven’t **alter**’ed the table schemas, you can use this file to reset the database to its original values (instance). If you have **alter**’ed the table schema, you will have to use the DDL+drop.sql file to reset the tables to their original schemas.
3. Now execute the query as you did with the DDL.sql file. You should get a successful response – ignore the “one row affected” message for now.
4. Click Edit->Clear Window in the main menu to clear the editor pane.
5. In the editor window, enter and execute the following -  
    **select count(1) from instructor**  
   You should get a result of 12 in the Data Output tab of the Output pane.
6. Close the Query Tool window.

Congratulations, you are done installing, configuring, and loading the database.