**POSTGRE SQL**

What is PostgreSQL?

PostgreSQL (pronounced as **post-gress-Q-L**) is an open source relational database management system (DBMS) developed by a worldwide team of volunteers. PostgreSQL is not controlled by any corporation or other private entity and the source code is available free of charge.

Key Features of PostgreSQL

PostgreSQL runs on all major operating systems, including Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), and Windows. It supports text, images, sounds, and video, and includes programming interfaces for C / C++, Java, Perl, Python, Ruby, Tcl and Open Database Connectivity (ODBC).

PostgreSQL supports a large part of the SQL standard and offers many modern features including the following −

* Complex SQL queries
* SQL Sub-selects
* Foreign keys
* Trigger
* Views
* Transactions
* Multiversion concurrency control (MVCC)
* Streaming Replication (as of 9.0)
* Hot Standby (as of 9.0)

You can check official documentation of PostgreSQL to understand the above-mentioned features. PostgreSQL can be extended by the user in many ways. For example by adding new −

* Data types
* Functions
* Operators
* Aggregate functions
* Index methods

Installing PostgreSQL on Linux/Unix

Follow the given steps to install PostgreSQL on your Linux machine. Make sure you are logged in as **root** before you proceed for the installation.

* Pick the version number of PostgreSQL you want and, as exactly as possible, the platform you want from EnterpriseDB
* I downloaded **postgresql-9.2.4-1-linux-x64.run** for my 64 bit CentOS-6 machine. Now, let us execute it as follows –

[root@host]# chmod +x postgresql-9.2.4-1-linux-x64.run

[root@host]# ./postgresql-9.2.4-1-linux-x64.run

------------------------------------------------------------------------

Welcome to the PostgreSQL Setup Wizard.

------------------------------------------------------------------------

Please specify the directory where PostgreSQL will be installed.

Installation Directory [/opt/PostgreSQL/9.2]:

* Once you launch the installer, it asks you a few basic questions like location of the installation, password of the user who will use database, port number, etc. So keep all of them at their default values except password, which you can provide password as per your choice. It will install PostgreSQL at your Linux machine and will display the following message –

Please wait while Setup installs PostgreSQL on your computer.

Installing

0% \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 50% \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 100%

#########################################

-----------------------------------------------------------------------

Setup has finished installing PostgreSQL on your computer.

* Follow the following post-installation steps to create your database –

[root@host]# su - postgres

Password:

bash-4.1$ createdb testdb

bash-4.1$ psql testdb

psql (8.4.13, server 9.2.4)

test=#

* You can start/restart postgres server in case it is not running using the following command –

[root@host]# service postgresql restart

Stopping postgresql service: [ OK ]

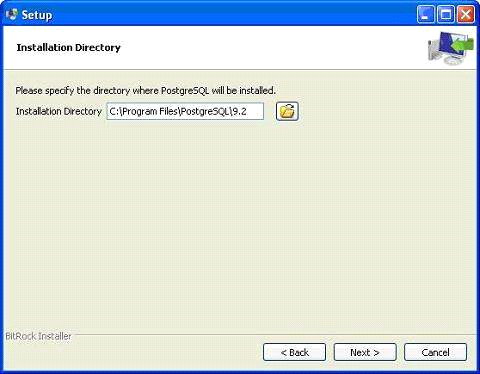
Starting postgresql service: [ OK ]

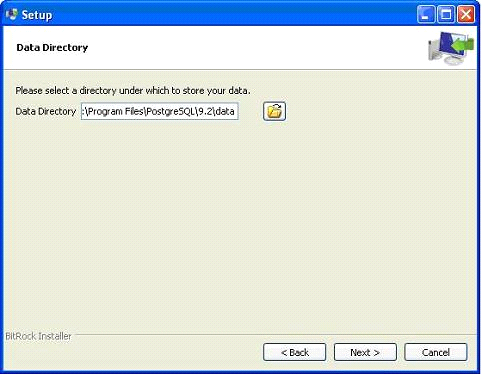
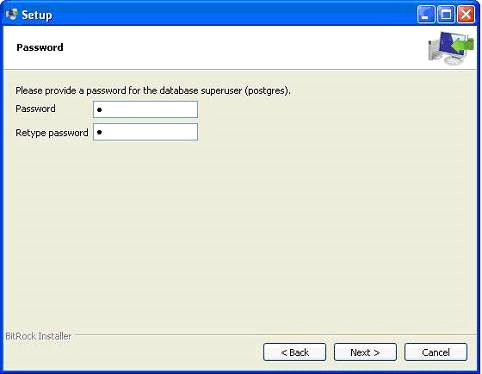
* If your installation was correct, you will have PotsgreSQL prompt **test=#** as shown above.

Installing PostgreSQL on Windows

Follow the given steps to install PostgreSQL on your Windows machine. Make sure you have turned Third Party Antivirus off while installing.

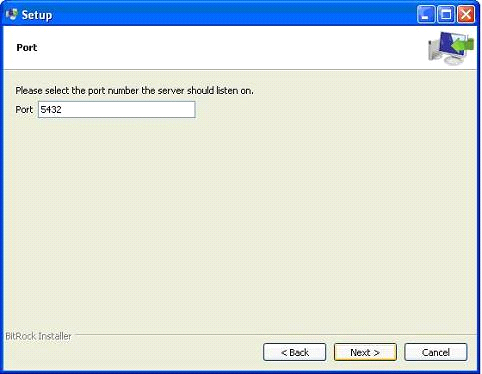
* Pick the version number of PostgreSQL you want and, as exactly as possible, the platform you want from EnterpriseDB
* I downloaded postgresql-9.2.4-1-windows.exe for my Windows PC running in 32bit mode, so let us run **postgresql-9.2.4-1-windows.exe** as administrator to install PostgreSQL. Select the location where you want to install it. By default, it is installed within Program Files folder.

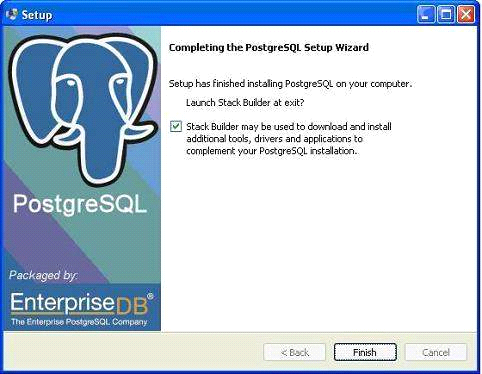


3.The next step of the installation process would be to select the directory where your data would be stored. By default, it is stored under the "data" directory.

* Next, the setup asks for password, so you

can use your favorite password.

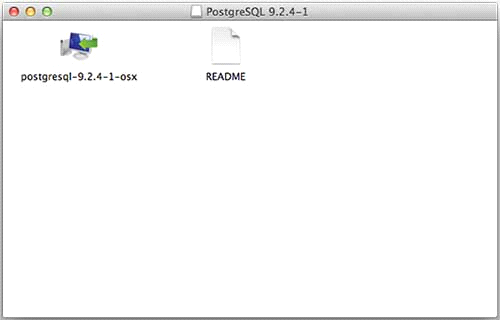


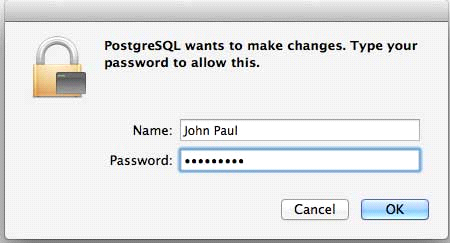
* The next step; keep the port as default.
* In the next step, when asked for "Locale", I selected "English, United States".
* It takes a while to install PostgreSQL on your system. On completion of the installation process, you will get the following screen. Uncheck the checkbox and click the Finish button.
* 
* After the installation process is completed, you can access pgAdmin III, StackBuilder and PostgreSQL shell from your Program Menu under PostgreSQL 9.2.

Installing PostgreSQL on Mac

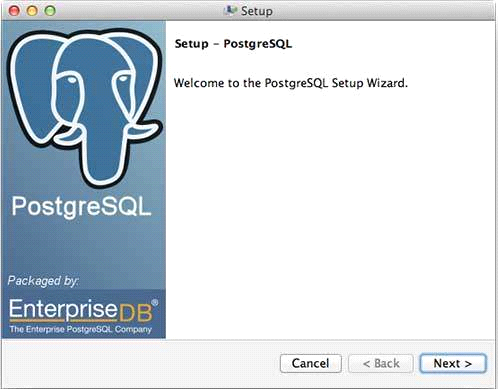
Follow the given steps to install PostgreSQL on your Mac machine. Make sure you are logged in as **administrator** before you proceed for the installation.

* Pick the latest version number of PostgreSQL for Mac OS available at EnterpriseDB
* I downloaded **postgresql-9.2.4-1-osx.dmg** for my Mac OS running with OS X version 10.8.3. Now, let us open the dmg image in finder and just double click it which will give you PostgreSQL installer in the following window –

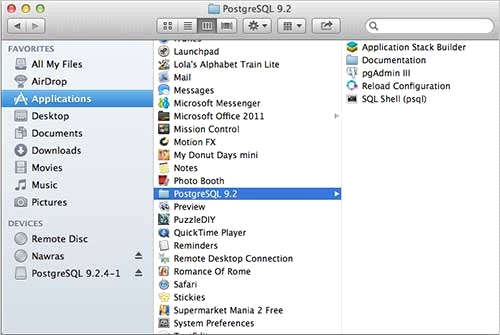


* Next, click the **postgres-9.2.4-1-osx** icon, which will give a warning message. Accept the warning and proceed for further installation. It will ask for the administrator password as seen in the following window −

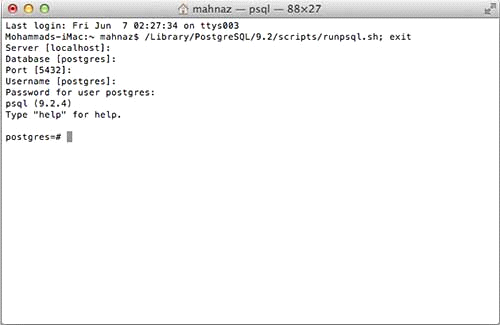
Enter the password, proceed for the installation, and after this step, restart your Mac machine. If you do not see the following window, start your installation once again.



* Once you launch the installer, it asks you a few basic questions like location
* of the installation, password of the user who will use database, port number etc. Therefore, keep all of them at their default values except the password, which you can provide as per your choice. It will install PostgreSQL in your Mac machine in the Application folder which you can check –



* Now, you can launch any of the program to start with. Let us start with SQL Shell. When you launch SQL Shell, just use all the default values it displays except, enter your password, which you had selected at the time of installation. If everything goes fine, then you will be inside postgres database and a **postgress#** prompt will be displayed as shown below −



Congratulations!!! Now you have your environment ready to start with PostgreSQL database programming.

The SQL Statement

An SQL statement is comprised of tokens where each token can represent either a keyword, identifier, quoted identifier, constant, or special character symbol. The table given below uses a simple SELECT statement to illustrate a basic, but complete, SQL statement and its components.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **SELECT** | **id, name** | **FROM** | **states** |
| Token Type | Keyword | Identifiers | Keyword | Identifier |
| Description | Command | Id and name columns | Clause | Table name |

1. Establish connection PSQL with Python

import psycopg2 as pg  
  
engine = pg.connect("dbname='beneficiary\_db' user='dbreaduser' host='3.7.81.85' port='5432' password='DBPROD@read$#$'")

2. Reading PSQL values with Python

import psycopg2 as pg  
import pandas as pd  
  
engine = pg.connect("dbname='beneficiary\_db' user='dbreaduser' host='3.7.81.85' port='5432' password='DBPROD@read$#$'")

df = pd.read\_sql\_query('select \* from beneficiary\_address limit 1', con=engine)  
  
print(df)

3. Load PSQL data to Local MysqlDatabase

import psycopg2 as pg  
import pandas as pd  
from sqlalchemy import create\_engine  
  
import pymysql  
  
from pandas.io import sql  
  
#connect\_to\_psql\_data  
engine = pg.connect("dbname='beneficiary\_db' user='dbreaduser' host='3.7.81.85' port='5432' password='DBPROD@read$#$'")

#query\_to\_retrive\_data

df = pd.read\_sql\_query('select \* from beneficiary\_address limit 1', con=engine)  
  
url = "mysql+pymysql://root:""@127.0.0.1/nb3"  
  
#engine = create\_engine("mysql+pymysql://" + "root" + ":" + "" + "@" + "127.0.0.1" + "/" + "beneficiary\_db")  
  
engine = create\_engine(url)  
df.to\_sql('beneficiary\_address', con = engine, if\_exists = 'replace')  
  
print("Done")

4.Load Data by ChunkSize

# Imports of Libraries

import psycopg2 as pg

import pandas as pd

from sqlalchemy import create\_engine

import pymysql

from pandas.io import sql

# 1. Migrate The Data from Postgrsql to Local MySQL Server in CHUNKS

# Establish Connection With Postgresql Server

engine = pg.connect("dbname='beneficiary\_db' user='dbreaduser' host='3.7.81.85' port='5432' password='DBPROD@read$#$'")

# Get The Data In Chunks of Size 100

df = pd.read\_sql\_query('select \* from beneficiary\_address', con=engine, chunksize=100)

# Push the DATA IN CHUNKS

for data in df:

# print(data)

# Establish Connection With Local Mysql Server

url = "mysql+pymysql://root:""@127.0.0.1/beneficiary\_db"

engine = create\_engine(url)

# Store The Data In Local MySQL

data.to\_sql('beneficiary\_address', con = engine, if\_exists = 'replace')

print("Done.")