

BT5110 Data Management and Warehousing

Tutorial 0 Special: Fundamental psql Operations

Mark Meng Huasong

School of Computing
National University of Singapore

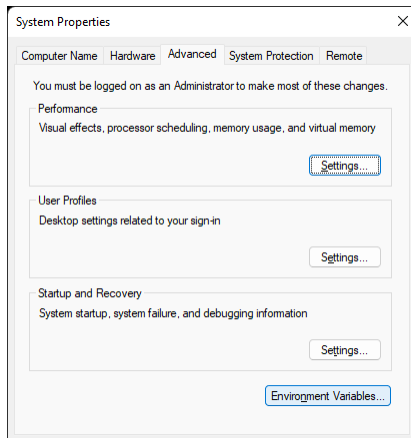
Aug 2021



Configuration on Windows PC (8/10/11)

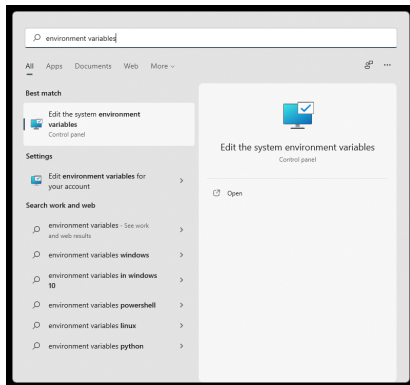
(1-a) Right click my PC, click “Properties”.

Go to “Advanced” tab, click the “Environment Variables” button.



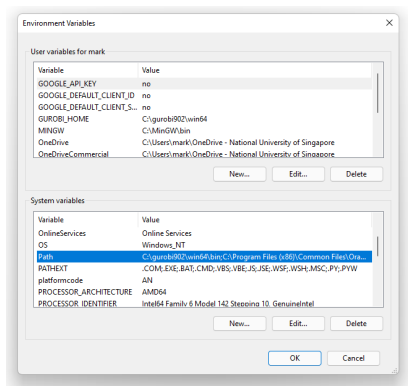
Configuration on Windows PC

(1-b) Alternatively, you may also click “start” and type keyword to search, like the figure below:



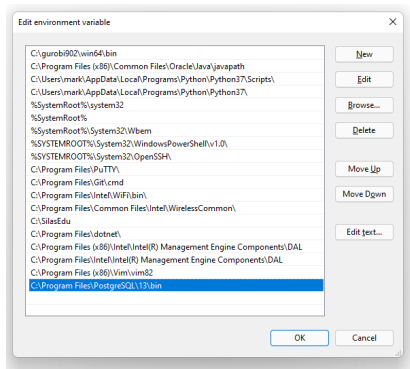
Configuration on Windows PC

(2) Find the “Path” in system variables.



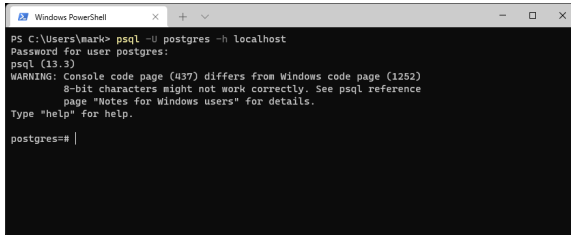
Configuration on Windows PC

(3) Add your PostgreSQL 13 installation path
(typically C:\\Program Files\\PostgreSQL\\13\\bin).



Connect to psql server

Open “Command Line”/“Windows Terminal”, and type in the command below to connect PostgreSQL.

A screenshot of a Windows PowerShell window. The title bar says "Windows PowerShell". The command prompt shows the user 'mark' at 'C:\Users\mark>' typing 'psql -U postgres -h localhost'. The prompt then asks for the password for the 'postgres' user. Below that, it shows 'psql (13.3)' and a warning message about console code page differences. Finally, it shows the prompt 'postgres=#' with a cursor.

```
PS C:\Users\mark> psql -U postgres -h localhost
Password for user postgres:
psql (13.3)
WARNING: Console code page (437) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

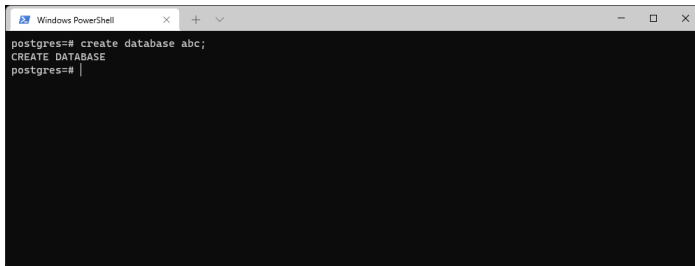
postgres=# |
```

Caution

Don't just enter “psql” as it assumes you are connecting with your Windows login account (e.g., **mark** in this case). However, in a fresh installed PostgreSQL, there is no such user but only “**postgres**”.

Create a new database

Enter “create database abc;” to **create** a database named “abc”.

A screenshot of a Windows PowerShell terminal window. The title bar at the top reads "Windows PowerShell" and includes standard window controls (minimize, maximize, close). The terminal content shows a PostgreSQL prompt "postgres=#" followed by the command "create database abc;". The command is executed, and the output "CREATE DATABASE" is displayed. The prompt "postgres=#" appears again with a vertical cursor line following it.

```
postgres=# create database abc;  
CREATE DATABASE  
postgres=# |
```

Browse all databases

Enter “\l” to **list** all databases.

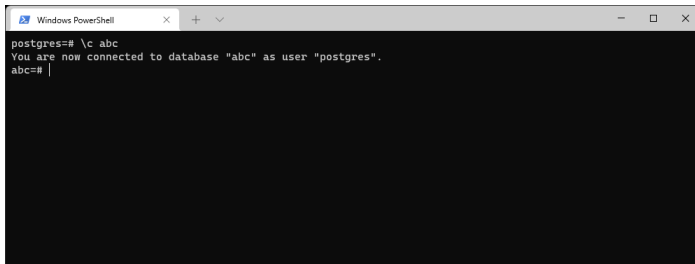
```
Windows PowerShell
postgres=# \l
```

List of databases					
Name	Owner	Encoding	Collate	Ctype	Access privileges
BT5110	postgres	UTF8	English_Singapore.1252	English_Singapore.1252	
abc	postgres	UTF8	English_Singapore.1252	English_Singapore.1252	
postgres	postgres	UTF8	English_Singapore.1252	English_Singapore.1252	
template0	postgres	UTF8	English_Singapore.1252	English_Singapore.1252	=c/postgres +
template1	postgres	UTF8	English_Singapore.1252	English_Singapore.1252	postgres=CTc/postgres =c/postgres

```
-- More --
```


Connect to a database

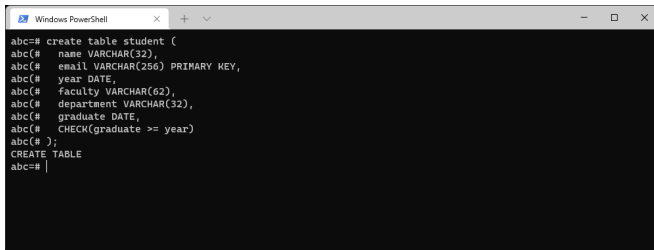
Enter “\c abc” to **connect** with the database “abc”.

A screenshot of a Windows PowerShell terminal window. The title bar shows 'Windows PowerShell' with standard window controls. The terminal has a black background with white text. The prompt 'postgres=#' is followed by the command '\c abc'. The output shows a successful connection: 'You are now connected to database "abc" as user "postgres".' followed by a new prompt 'abc=#' with a cursor.

```
postgres=# \c abc
You are now connected to database "abc" as user "postgres".
abc=# |
```

Create a table

Let's create a table "student", which is exactly same as the one we will use for Tutorial 1.

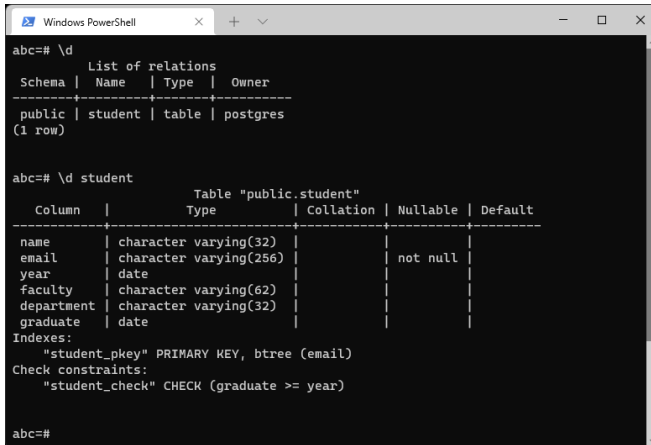
A screenshot of a Windows PowerShell window. The title bar shows 'Windows PowerShell' with standard window controls. The command prompt shows a series of commands to create a table named 'student'. The commands are: 'abc=# create table student (' followed by several lines of column definitions: 'abc(# name VARCHAR(32),', 'abc(# email VARCHAR(256) PRIMARY KEY,', 'abc(# year DATE,', 'abc(# faculty VARCHAR(62),', 'abc(# department VARCHAR(32),', 'abc(# graduate DATE,', and 'abc(# CHECK(graduate >= year)'. This is followed by 'abc(#);' and 'CREATE TABLE'. The prompt 'abc=# |' is at the end of the last line.

```
abc=# create table student (  
abc(#   name VARCHAR(32),  
abc(#   email VARCHAR(256) PRIMARY KEY,  
abc(#   year DATE,  
abc(#   faculty VARCHAR(62),  
abc(#   department VARCHAR(32),  
abc(#   graduate DATE,  
abc(#   CHECK(graduate >= year)  
abc(# );  
CREATE TABLE  
abc=# |
```

View existing tables

Enter “\d” to **display** all relations in the current database.

Enter “\d student” to **display** the details of the table “student”.



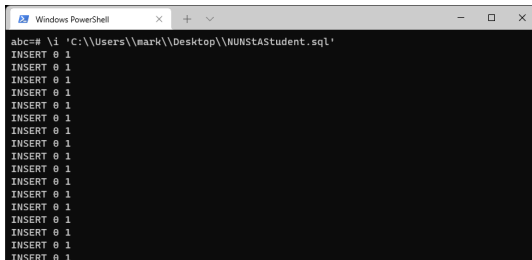
```
abc=# \d
          List of relations
 Schema | Name   | Type  | Owner
-----+-----+-----+-----
 public | student | table | postgres
(1 row)

abc=# \d student
          Table "public.student"
   Column   |          Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
 name       | character varying(32)  |           |           |
 email      | character varying(256) |           | not null  |
 year       | date                   |           |           |
 faculty    | character varying(62)  |           |           |
 department | character varying(32)  |           |           |
 graduate   | date                   |           |           |
Indexes:
    "student_pkey" PRIMARY KEY, btree (email)
Check constraints:
    "student_check" CHECK (graduate >= year)

abc=#
```

Execute an SQL script from files

Now let's insert some data. To save time, we just make use of the SQL script provided for our tutorials – “NUNStASStudent.sql”

A screenshot of a Windows PowerShell window. The title bar reads "Windows PowerShell". The command prompt shows the execution of a file: `abc=# .i 'C:\\Users\\mark\\Desktop\\NUNStASStudent.sql'`. Below this, there is a list of 15 `INSERT 0 1` statements, indicating that 15 rows of data were successfully inserted into the database.

```
Windows PowerShell
abc=# .i 'C:\\Users\\mark\\Desktop\\NUNStASStudent.sql'
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
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
```

Caution

Please be cautious that the file path on Windows must be formatted with double ‘//’ and enclosed within single-quote symbols (“C:\\Users\\mark\\Desktop\\NUNStASStudent.sql” in this case). Otherwise a Permission Denied error will be returned.

Execute an SQL script manually

Enter SQL query manually to view outputs.

```
Windows PowerShell
abc=# select * from student;
```

name	email	year	faculty	department	graduate
XIE XIN	xiexin2011@gmail.com	2007-01-01	Faculty of Science	Chemistry	
HUANG RAN	huangran1991@yahoo.com	2007-08-01	Faculty of Science	Biology	
GOH ENG CHYE	gohengchye1992@msn.com	2007-08-01	School of Computing	CS	
GOH HUI YING	gohhuiying1989@gmail.com	2008-01-01	Faculty of Science	Biology	
FANG HAN	fanghan2011@hotmail.com	2008-01-01	Faculty of Arts and Social Science	Geography	
DING KUAN CHONG	dingkuanchong2010@msn.com	2008-08-01	Faculty of Engineering	CE	
TAY WEI GUO	tayweiguo1989@msn.com	2010-01-01	Faculty of Engineering	CE	
ONG KAH HONG	ongkahhong1991@gmail.com	2008-01-01	Faculty of Science	Math	
PENG JIAYUAN	pengjiayuan2011@hotmail.com	2008-01-01	Faculty of Science	Biology	
HUANG ZHANPENG	huangzhanpeng1992@msn.com	2010-01-01	Faculty of Arts and Social Science	Geography	

```
-- More --
```

```
Windows PowerShell
abc=# select COUNT(*) from student;
```

count
103

```
(1 row)

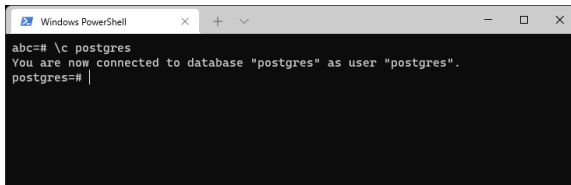
abc=#
```

Disconnect & Drop a database

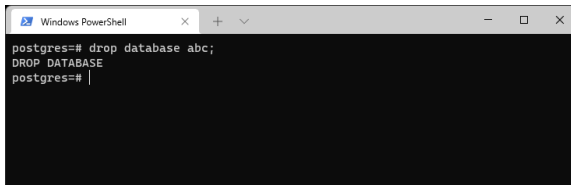
To drop the current database, you have to disconnect it first.

To do that, just connect to the other one by entering “\c postgres” to **connect** to the default database “postgres”.

Then enter “drop database abc” to **drop** the database we have just created.



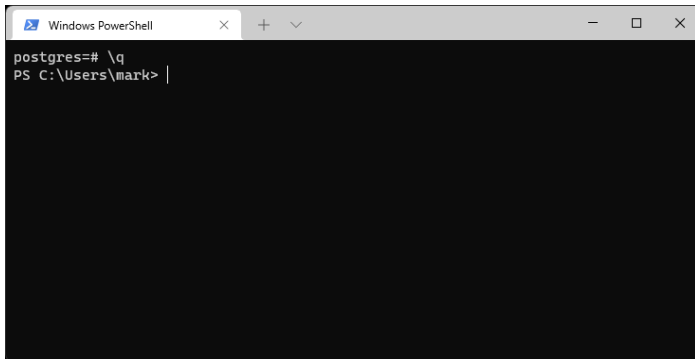
```
Windows PowerShell
abc=# \c postgres
You are now connected to database "postgres" as user "postgres".
postgres=#
```



```
Windows PowerShell
postgres=# drop database abc;
DROP DATABASE
postgres=#
```

Quit the psql

Enter “\q” to **quit**.

A screenshot of a Windows PowerShell terminal window. The title bar at the top reads "Windows PowerShell" and includes standard window controls (minimize, maximize, close). The terminal content shows a prompt "postgres=#" followed by the command "\q" entered on the same line. The next line shows the prompt "PS C:\Users\mark>" with a cursor at the end, indicating the command has been executed and the user has returned to the PowerShell prompt.

```
Windows PowerShell
postgres=# \q
PS C:\Users\mark> |
```

For any further question, please feel free to email me:

hmeng@comp.nus.edu.sg