

# Data Base System - Hw3

## **Tutorial & Submission Rules**

**TA: Tzu-Heng Huang & Zheng-Xian Cai**

**TA Hour: Thur. & Fri. 14 ~ 16 @DMLAB**

**Email: [zihenghi@gmail.com](mailto:zihenghi@gmail.com) & [p154968@gmail.com](mailto:p154968@gmail.com)**

**Course Instructor: Man-Kwan Shan**



# Content

1. About Hw3
2. Download PostgresDB
3. Create Database
4. Create Table
5. Import your Data
6. Check your Data
7. Design and Implement SQL Query
8. Save your Query
9. Submission Rules



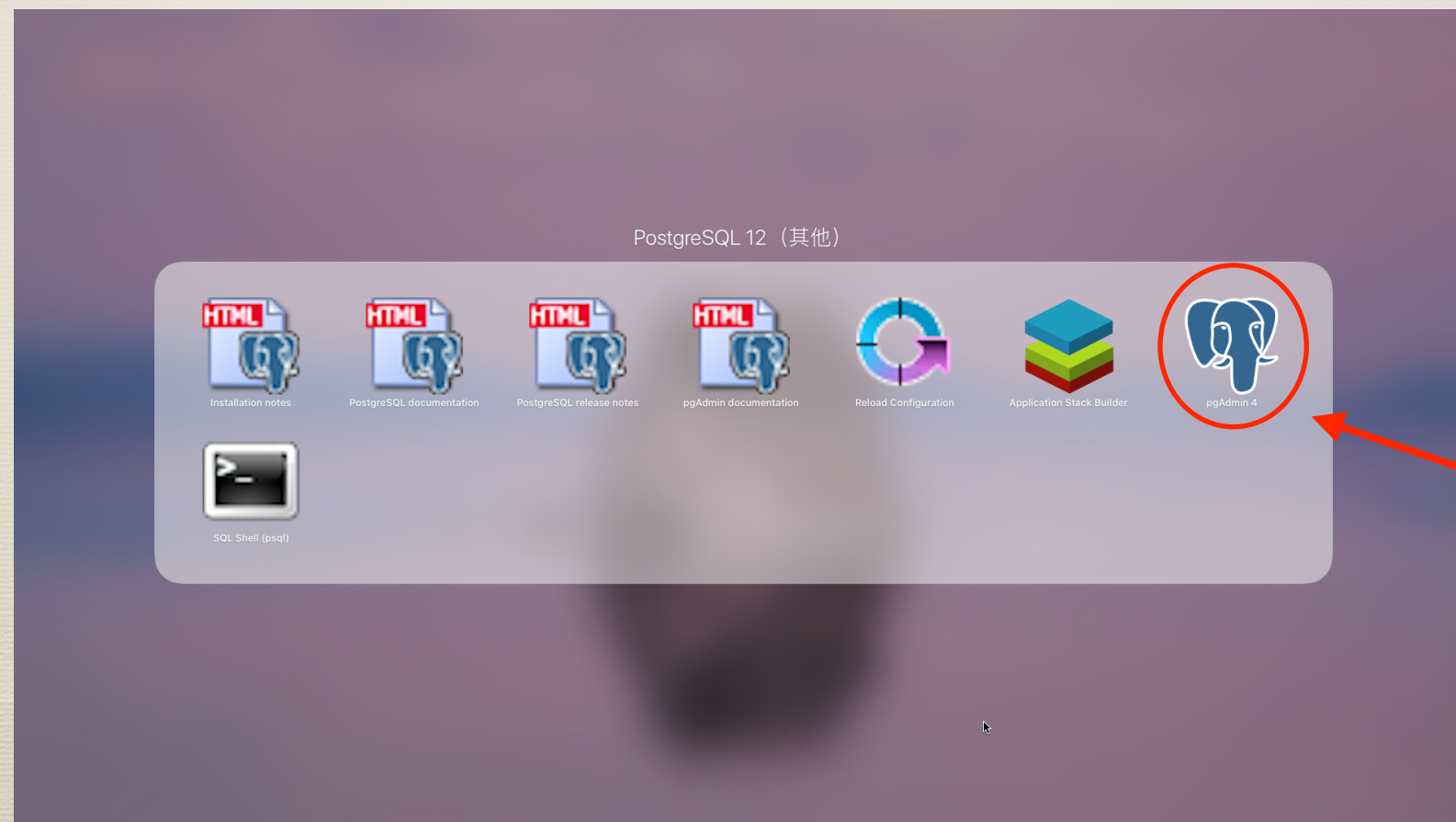
# 1. About Hw3

- This homework is to construct a set of SQL queries for analyzing a dataset. For this, you will look into IMDB data which contains 6 tables.
- Six files in csv format are provided which have to be imported into database.
- In this HW, you need to
  - create database schema.
  - import data.
  - design and implement SQL query.
- This homework is an opportunity to:
  - (1) learn basic and certain advanced SQL features.
  - (2) get familiar with using a full-featured DBMS, **PostgresDB**, that can be useful for you in the future.
- We will check your query results and compare them via “diff” command.
- Please enjoy!



## 2. Download PostgresDB

- What is PostgresDB?  
-> Features of PostgresDB
- Where to download?  
-> Database Download
- Launch your pgAdmin, a management tool for PostgresDB!





# Launch Successfully

Internet | Python | Python | How to | GAM: T | Particul | pmBro | Particul | Read th | Call for | Spatial | Discov | Globec | Organiz | CaaS: C | Instead | Use al | 知 关于本 | 5. 定義 | Postgre | pg/ x | HomeW | What is | +

127.0.0.1:61350/browser/

應用程式 | News | Programing | Self Study | English | Argonne | Sinica | Job Application | Others | Graph


pgAdmin File Object Tools Help

Browser Dashboard Properties SQL Statistics Dependencies Dependents

Servers (1)

- PostgreSQL 12
  - Databases (3)
    - DB\_Test
    - IMDB
    - postgres
  - Login/Group Roles
  - Tablespaces


Welcome


 **pgAdmin**  
Management Tools for PostgreSQL

Feature rich | Maximises PostgreSQL | Open Source


pgAdmin is an Open Source administration and management tool for the PostgreSQL database. It includes a graphical administration interface, an SQL query tool, a procedural code debugger and much more. The tool is designed to answer the needs of developers, DBAs and system administrators alike.


Quick Links


 Add New Server


 Configure pgAdmin

Getting Started

 PostgreSQL Documentation

 pgAdmin Website

 Planet PostgreSQL

 Community Support

# 3. Create Database (for example)

The screenshot shows the pgAdmin 4 web interface. On the left, the 'Databases (3)' folder is highlighted with a red circle and an arrow pointing to the text 'Right Click and "Create"'. In the center, the 'Create - Database' dialog box is open. The 'Database' field is highlighted with a red circle and an arrow pointing to the text 'Name your DB'. The 'Save' button is highlighted with a red circle and an arrow pointing to the text 'Save it'.

**Right Click and "Create"**

**Name your DB**

**Save it**



# 4. Create Table (for example)

The screenshot shows the pgAdmin 4 web interface. On the left, the 'Servers' tree is expanded to 'Databases (3)', and a right-click context menu is open with 'Query Tool...' selected. The main panel shows the 'Query Editor' with the following SQL code:

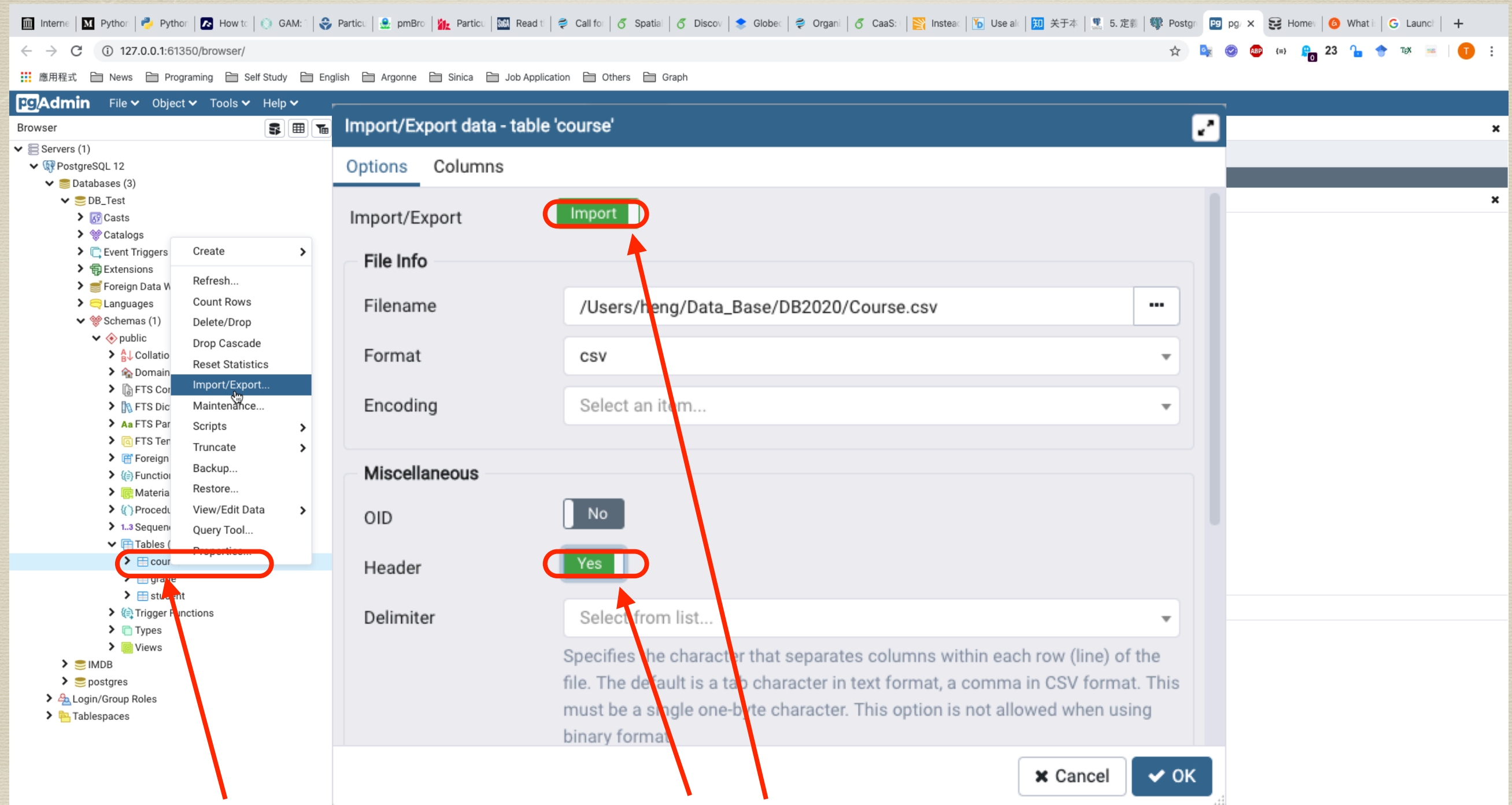
```
1 CREATE TABLE student(  
2     student_id character varying PRIMARY KEY,  
3     student_name character varying NOT NULL,  
4     gender character varying NOT NULL,  
5     birthday date NOT NULL,  
6     fruit character varying NOT NULL  
7 );  
8  
9 CREATE TABLE course(  
10     course_id character varying PRIMARY KEY,  
11     course_name character varying NOT NULL,  
12     credit INT NOT NULL  
13 );  
14  
15 CREATE TABLE grade(  
16     student_id character varying REFERENCES student(student_id),  
17     course_id character varying REFERENCES course(course_id),  
18     score INT NOT NULL  
19 );
```

At the bottom of the interface, there are three red arrows pointing to specific elements:

- An arrow points from the text "Right Click and 'Query Tool'" to the 'Query Tool...' option in the context menu.
- An arrow points from the text "Create Table" to the SQL code in the Query Editor.
- An arrow points from the text "Execute Program" to the 'Execute Program' button (a play icon) in the toolbar.

Here is the code. (./example/create\_table.sql)

# 5. Import Data (for example)



Find your table and “Import Data”

“Import” and “Header” switch on

Take our template as example, you can import Course.csv, Student.csv and Grade.csv.



# Import Data

Import/Export data - table 'akas'

Options Columns

OID ☐ No

Header ☒ Yes

Delimiter   
Specifies the character that separates columns within each row (line) of the file. The default is a tab character in text format, a comma in CSV format. This must be a single one-byte character. This option is not allowed when using binary format.

Quote   
Specifies the quoting character to be used when a data value is quoted. The default is double-quote. This must be a single one-byte character. This option is allowed only when using CSV format.

Escape   
Specifies the character that should appear before a data character that matches the QUOTE value. The default is the same as the QUOTE value (so that the quoting character is doubled if it appears in the data). This must be a single one-byte character. This option is allowed only when using CSV format.

Cancel OK

Note that, in IMDB dataset, you have change the default Escape option to “double quotation marks”.



# 6. Check your Data (for example)

The screenshot shows the pgAdmin 4 web interface. On the left, the 'course' table is selected in the 'Tables (3)' folder under the 'public' schema. A context menu is open, and the 'View/Edit Data' option is highlighted, with a sub-menu showing 'All Rows' selected. A red arrow points from the text 'View your Data' to the 'course' table. The 'Query Editor' shows a SQL query: `SELECT * FROM public.course`. The 'Data Output' tab displays the following data:

	course_id [PK] character varying	course_name character varying	credit integer
1	CS50	Data Base System	3
2	CS51	Algorithm Analysis	3
3	CS52	Deep Learning	3
4	CS53	Web Search Mining	3
5	CS54	Graph Theory	3

Correct!

View your Data



# 7. Design and Implement SQL Query (for example)

The screenshot shows the pgAdmin web interface. On the left, a tree view shows the database structure. The main area is divided into two panes. The top pane, labeled 'Query Editor', contains an SQL query. The bottom pane, labeled 'Data Output', shows the results of the query in a table format. A red arrow points from the query to the results table. A green status bar at the bottom right indicates the query was successfully run.

**Query Editor**

```
SELECT student_name, grade.*
FROM student, grade
WHERE student_name = 'Brian' and student.student_id = grade.student_id
order by grade.score
```

**Data Output**

	student_name	student_id	course_id	score
1	Brian	1000	CS51	70
2	Brian	1000	CS50	86
3	Brian	1000	CS52	91

Successfully run. Total query runtime: 93 msec. 3 rows affected.

Query & Result

Ref: SQL Language



# 8. Save your Query (for example)

The screenshot shows the pgAdmin 4 web interface. The left sidebar displays the database structure, including 'Servers (1)' with 'PostgreSQL 12', and 'Databases (3)' with 'DB\_Test'. The main panel shows the 'Query Editor' with a SQL query:

```
1 SELECT student_name, grade.*
2 FROM student_grade
3 WHERE student_name = 'Brian' and student.student_id = grade.student_id
4 order by grade.score
```

A red arrow points to the 'Save' icon (a floppy disk) in the toolbar, with the text 'Save as a query.sql file'.

The 'Data Output' tab is selected, showing the result set:

	student_name character varying	student_id character varying	course_id character varying	score integer
1	Brian	1000	CS51	70
2	Brian	1000	CS50	86
3	Brian	1000	CS52	91

A green notification bar at the bottom right states: 'Successfully run. Total query runtime: 93 msec. 3 rows affected.'

Ref: SQL Language



## 9. Submission Rules (for HW)

- In your HW package, we have already prepared this document, a template file and a IMDB file for you.

```
[23:37 Heng [~/Data_Base/DB2020/Hw3] $ tree
```

```
.
├── Hw3\ Tutorial.pdf
├── IMDB
│   ├── create_table.sql
│   └── data.zip
└── example
    ├── Course.csv
    ├── Grade.csv
    ├── Student.csv
    ├── create_table.sql
    └── query.sql
```

```
2 directories, 8 files
```

File Structure

```
[20:44 Heng [~/Data_Base/DB2020/solution] $ tree
```

```
.
├── q1.sql
├── q10.sql
├── q2.sql
├── q3.sql
├── q4.sql
├── q5.sql
├── q6.sql
├── q7.sql
├── q8.sql
└── q9.sql
```

```
0 directories, 10 files
```

Submission Format

- Write down your SQL query for each question and save as sql files named by their question number. (q1.sql, q2.sql, q3.sql, and so on).
- Compress ten sql files to a zip file named in “**your student ID.zip**” and submit to WM5.