Data Base System - Hw3

Tutorial & Submission Rules

TA: Tzu-Heng Huang & Zheng-Xian Cai
TA Hour: Thur. & Fri. 14 ~ 16 @DMLAB
Email: zihengh1@gmail.com & p154968@gmail.com
Course Instructor: Man-Kwan Shan

Content

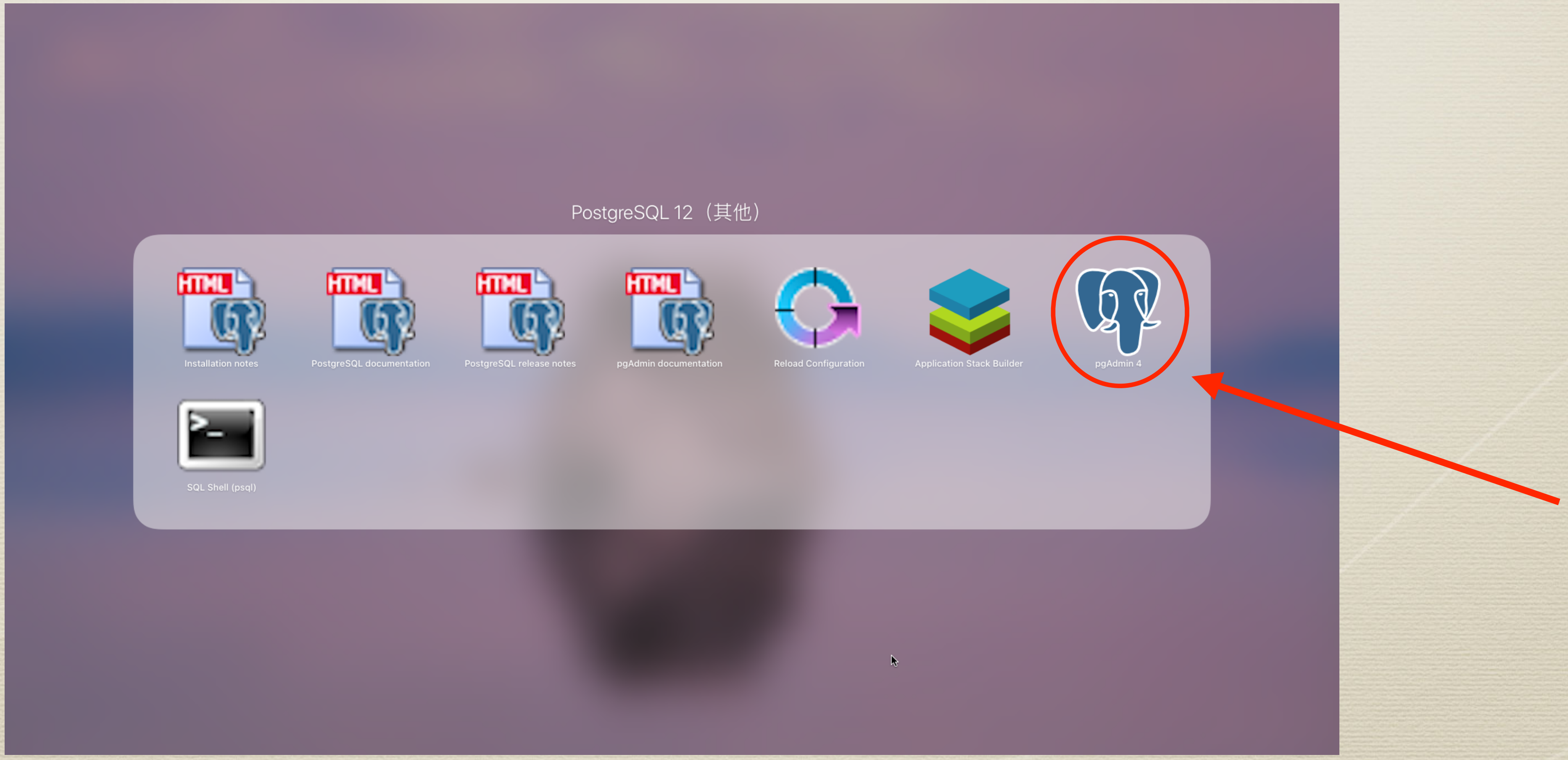
1. About Hw3
2. Download PostgresDB
3. Create Database
4. Create Table
5. Import your Data
6. Check your Data
7. Design and Implement SQL Query
8. Save your Query
9. Submission Rules

1. About Hw3

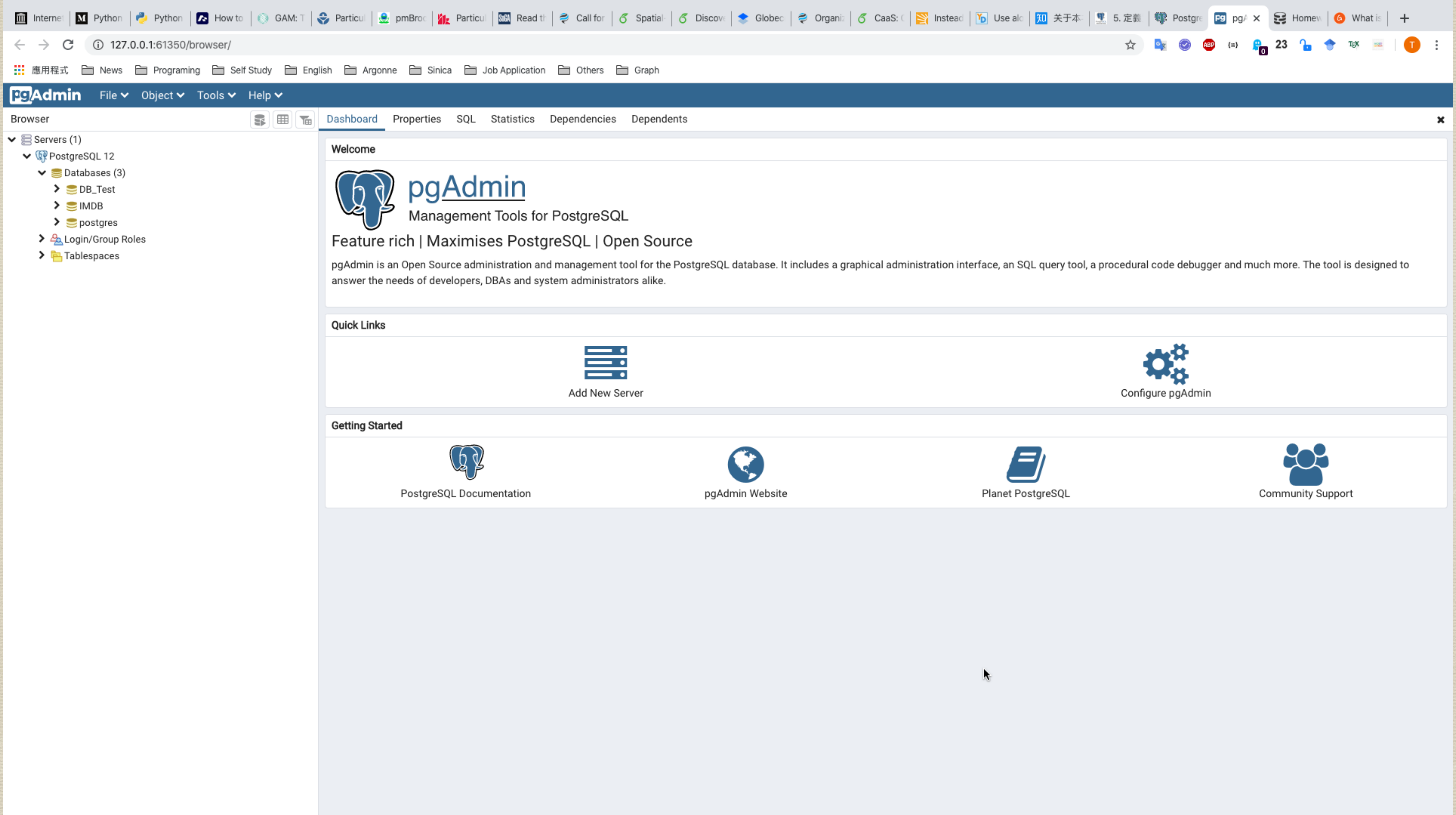
- This homework is to construct a set of SQL queries for analyzing a dataset. For this, you will look into IMDB data which contains 6 tables.
- Six files in csv format are provided which have to be imported into database.
- In this HW, you need to
  - create database schema.
  - import data.
  - design 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



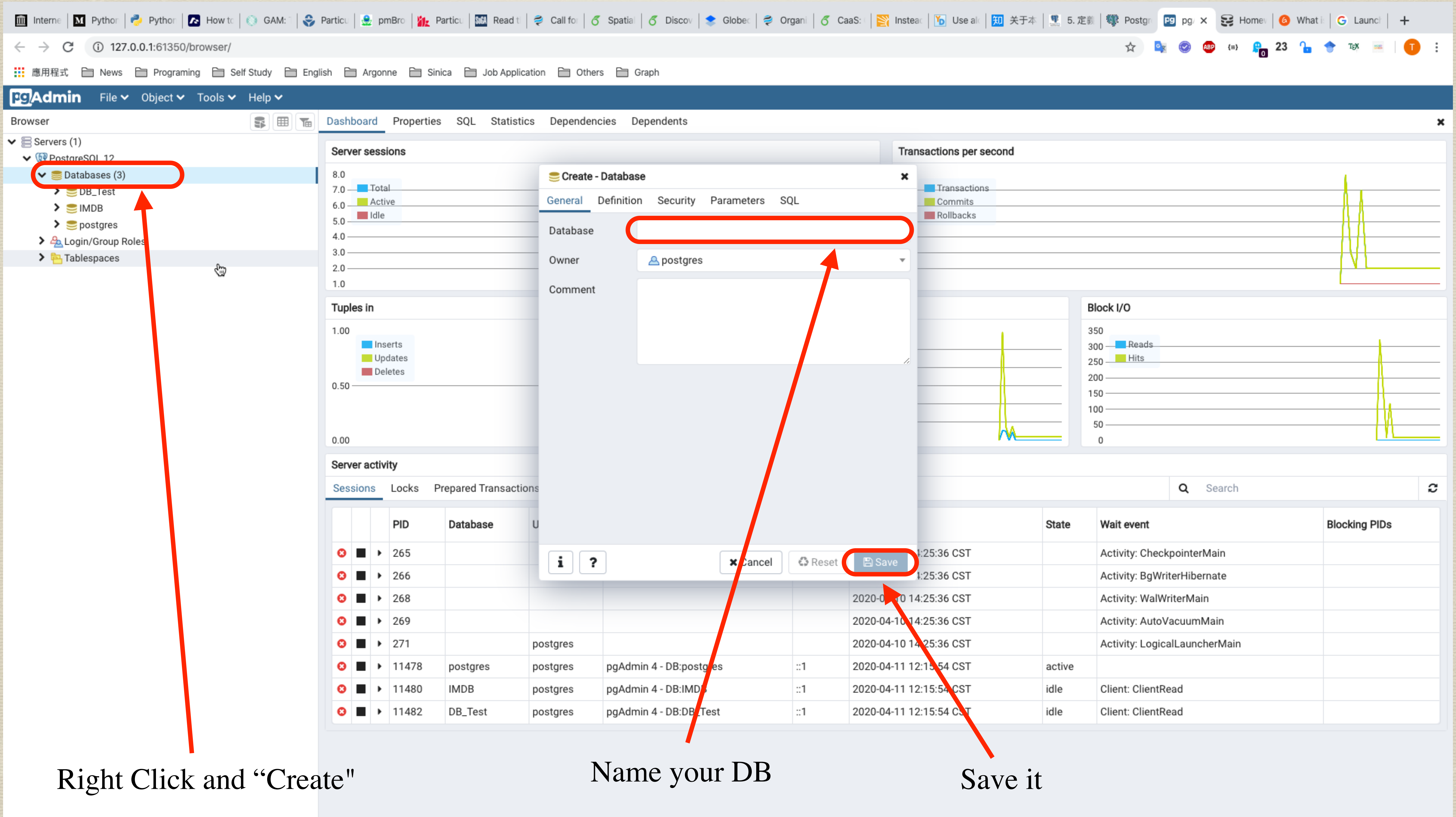
pgAdmin

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.

3. Create Database (for example)



Create - Database

Database:

Owner: postgres

Comment:

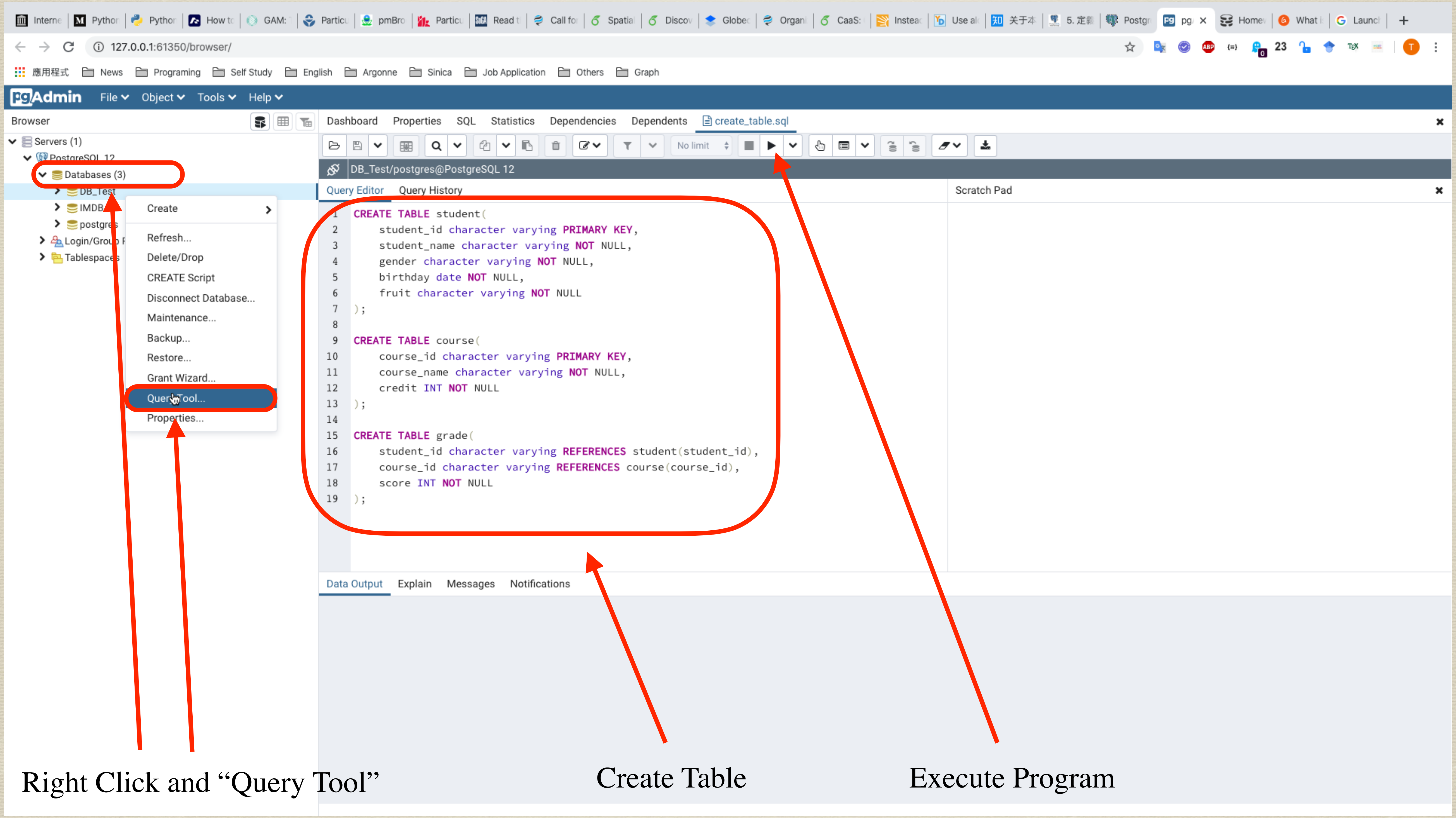
	PID	Database	U				State	Wait event	Blocking PIDs
	265					:25:36 CST		Activity: CheckpointerMain	
	266					:25:36 CST		Activity: BgWriterHibernate	
	268					2020-04-10 14:25:36 CST		Activity: WalWriterMain	
	269					2020-04-10 14:25:36 CST		Activity: AutoVacuumMain	
	271		postgres			2020-04-10 14:25:36 CST		Activity: LogicalLauncherMain	
	11478	postgres	postgres	pgAdmin 4 - DB:postgres	::1	2020-04-11 12:15:54 CST	active		
	11480	IMDB	postgres	pgAdmin 4 - DB:IMDB	::1	2020-04-11 12:15:54 CST	idle	Client: ClientRead	
	11482	DB_Test	postgres	pgAdmin 4 - DB:DB_Test	::1	2020-04-11 12:15:54 CST	idle	Client: ClientRead	

Right Click and "Create"

Name your DB

Save it

4. Create Table (for example)



```sql
CREATE TABLE student(
    student_id character varying PRIMARY KEY,
    student_name character varying NOT NULL,
    gender character varying NOT NULL,
    birthday date NOT NULL,
    fruit character varying NOT NULL
);

CREATE TABLE course(
    course_id character varying PRIMARY KEY,
    course_name character varying NOT NULL,
    credit INT NOT NULL
);

CREATE TABLE grade(
    student_id character varying REFERENCES student(student_id),
    course_id character varying REFERENCES course(course_id),
    score INT NOT NULL
);
```

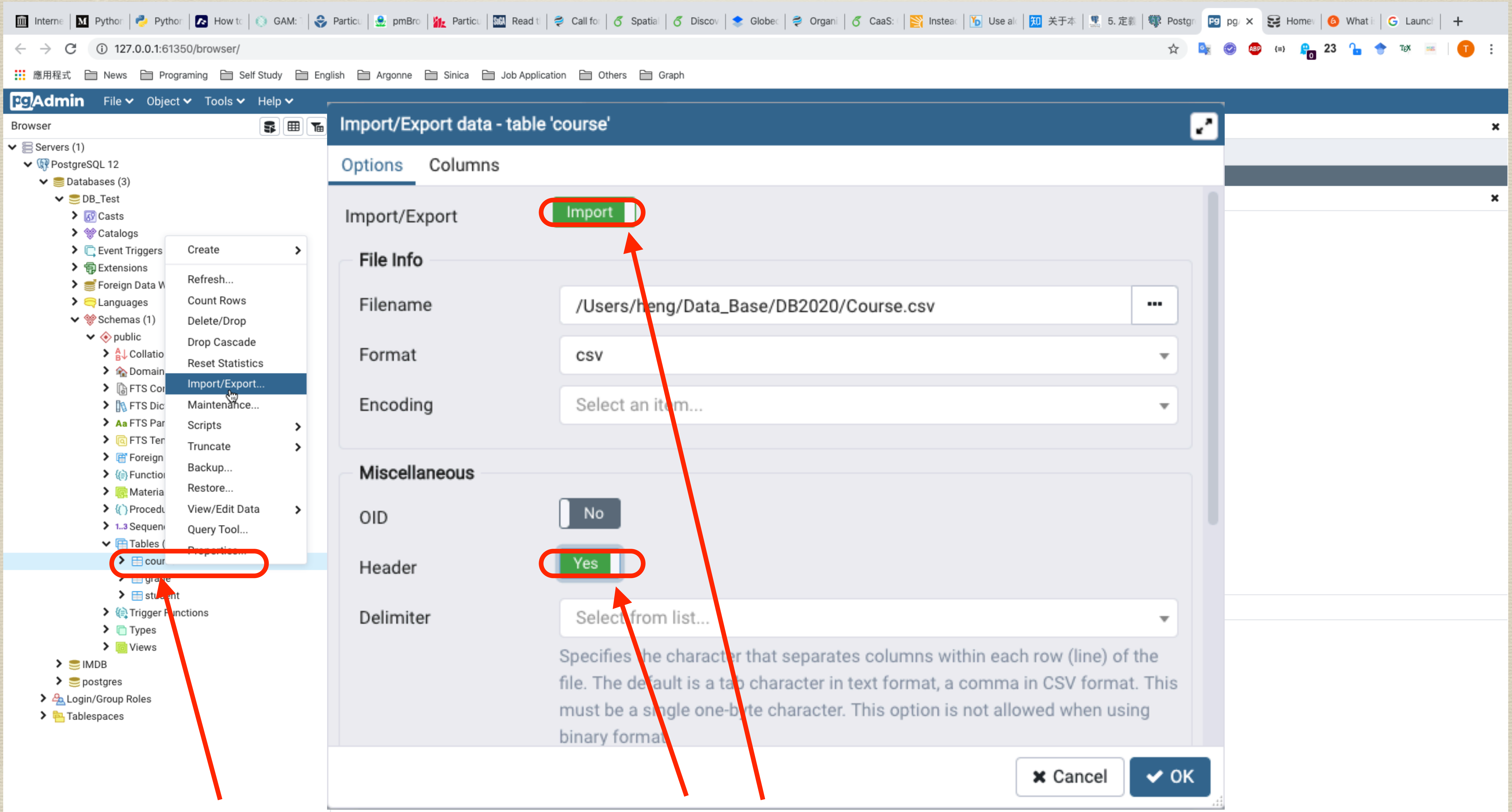
Right Click and “Query Tool”

Create Table

Execute Program

Here is the code. (./example/create_table.sql)

5. Import Data (for example)



Import/Export data - table 'course'

Import/Export: Import

File Info

Filename: /Users/heng/Data_Base/DB2020/Course.csv

Format: csv

Encoding: Select an item...

Miscellaneous

OID: No

Header: Yes

Delimiter: Select from list...

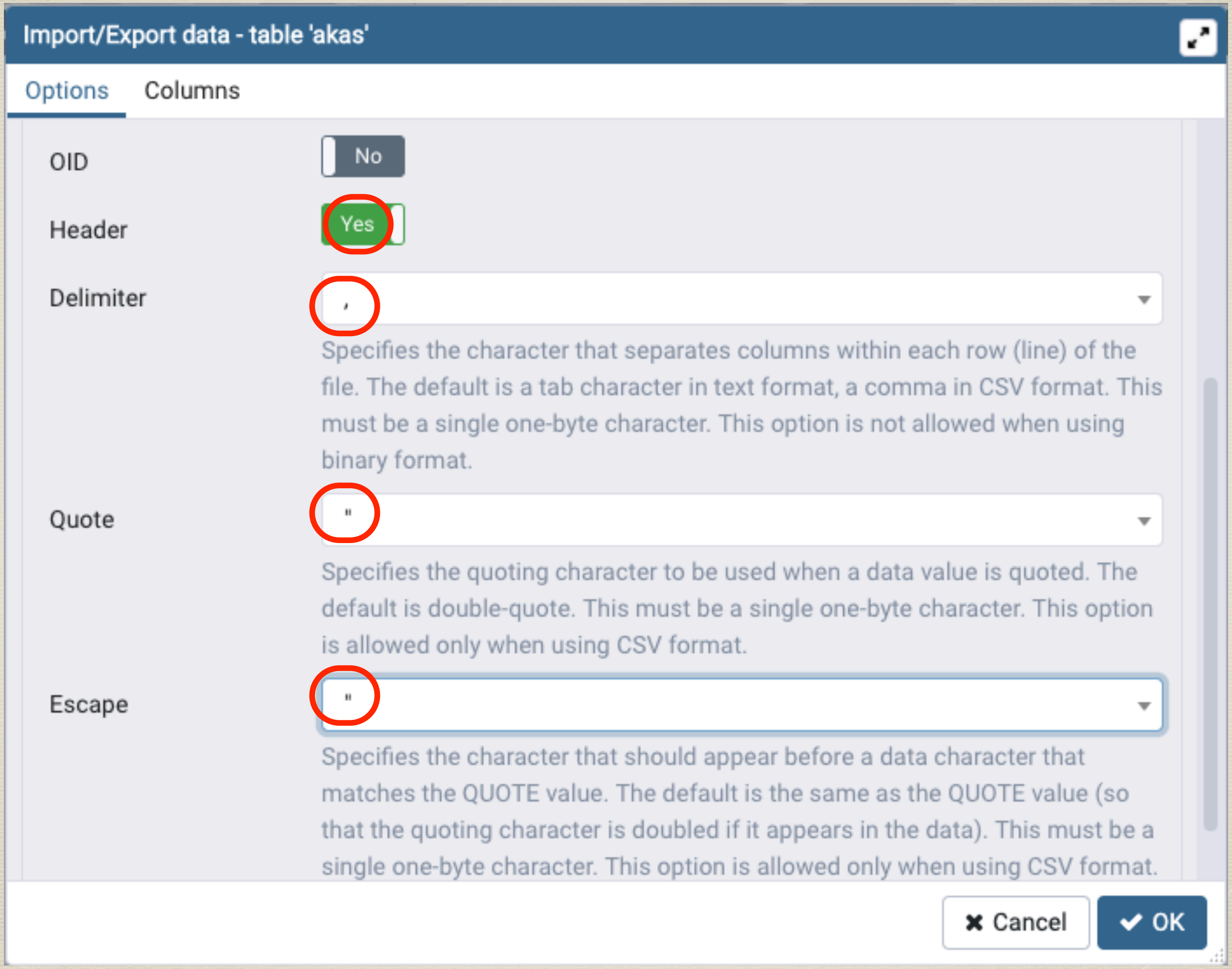
Specifies the character that separates columns within each row (line) of the file. The default is a tab character in text format, a comma in CSV format. This must be a single one-byte character. This option is not allowed when using binary format

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'

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: "

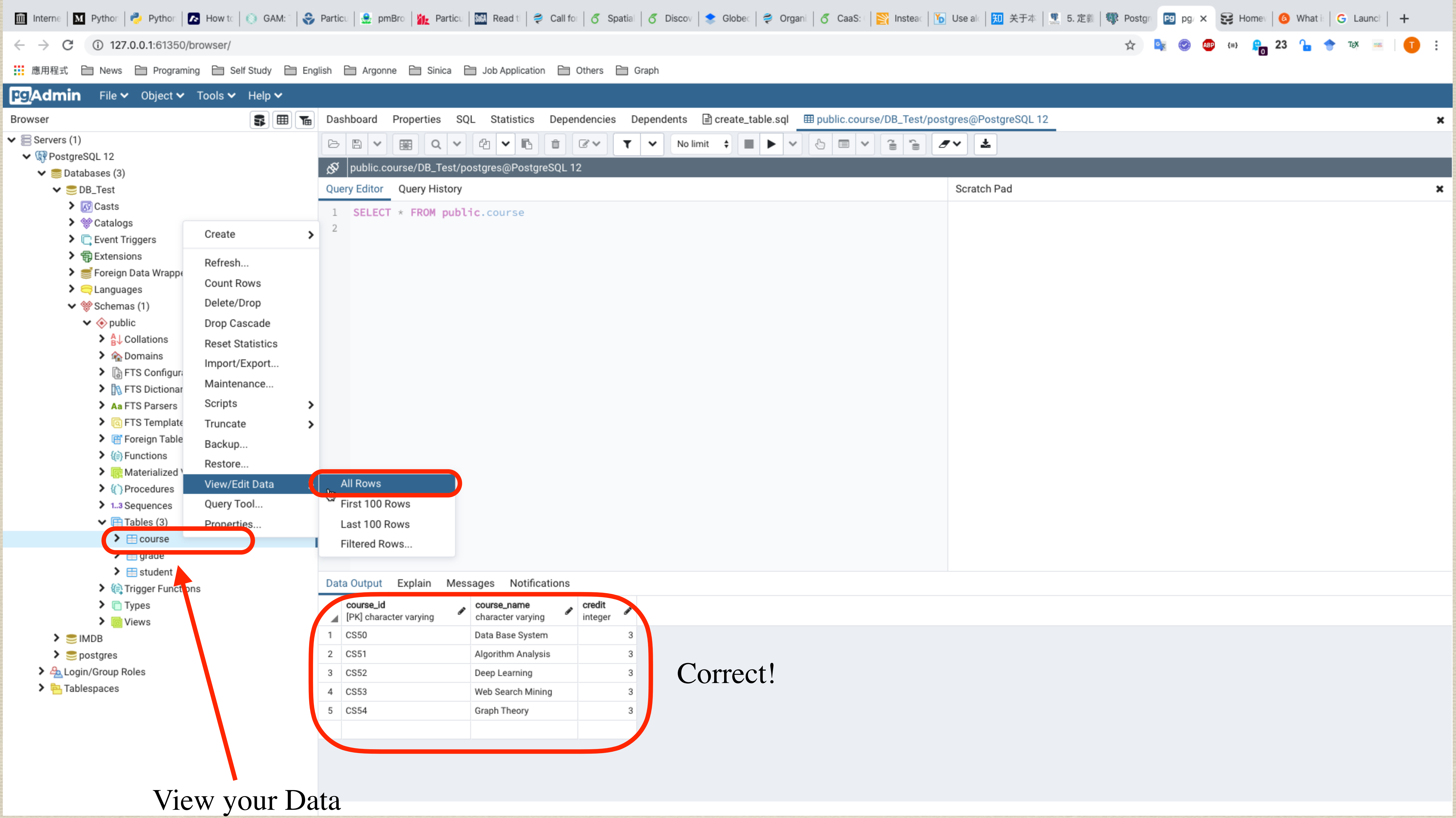
Specifies the quoting character to be used when a data value is quoted. The default is double-quote. This must be a single one-byte character. This option is allowed only when using CSV format.

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.

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

6. Check your Data (for example)



127.0.0.1:61350/browser/

public.course/DB_Test/postgres@PostgreSQL 12

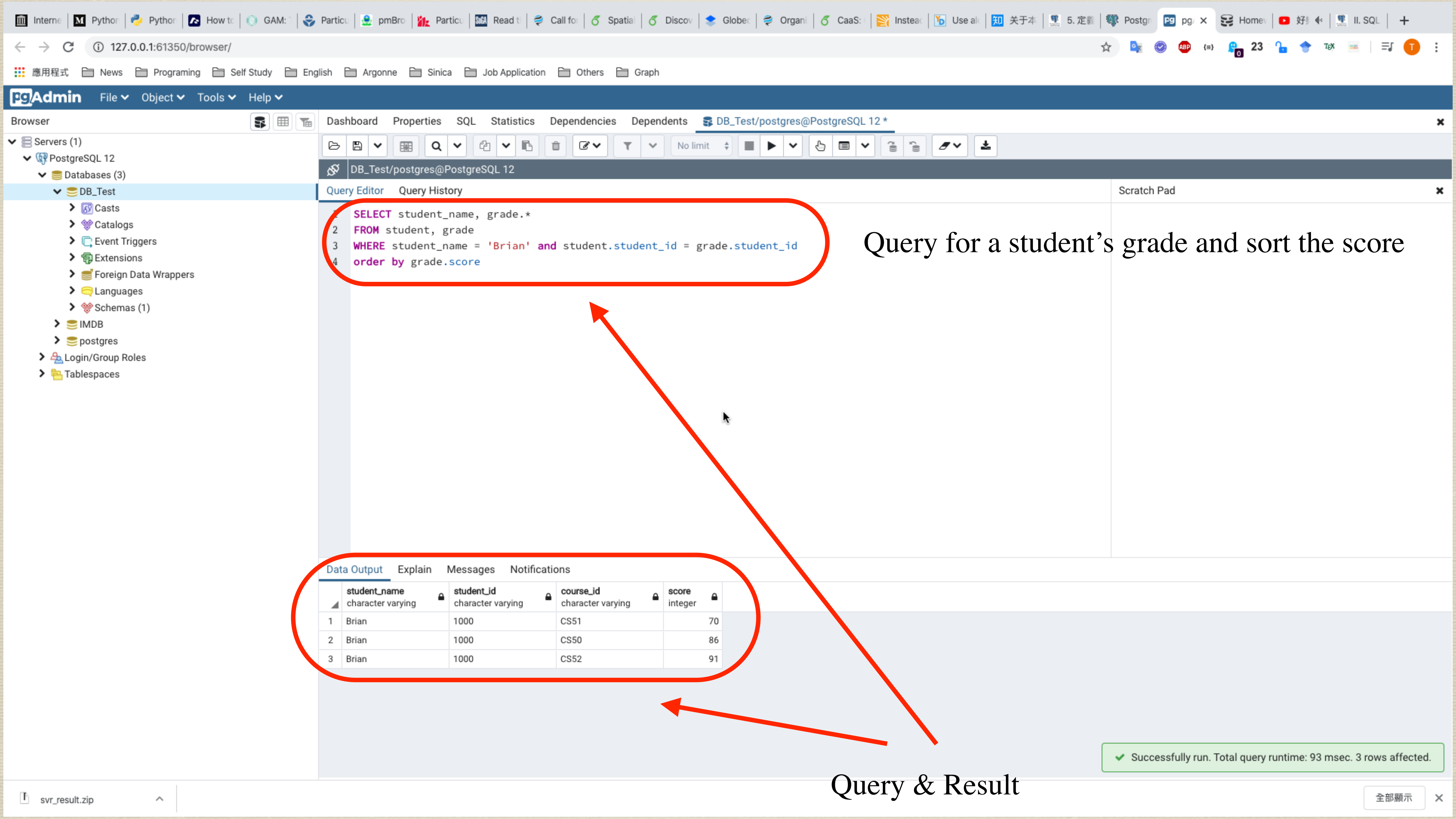
```
SELECT * FROM public.course
```

	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)



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

Query for a student’s grade and sort the score

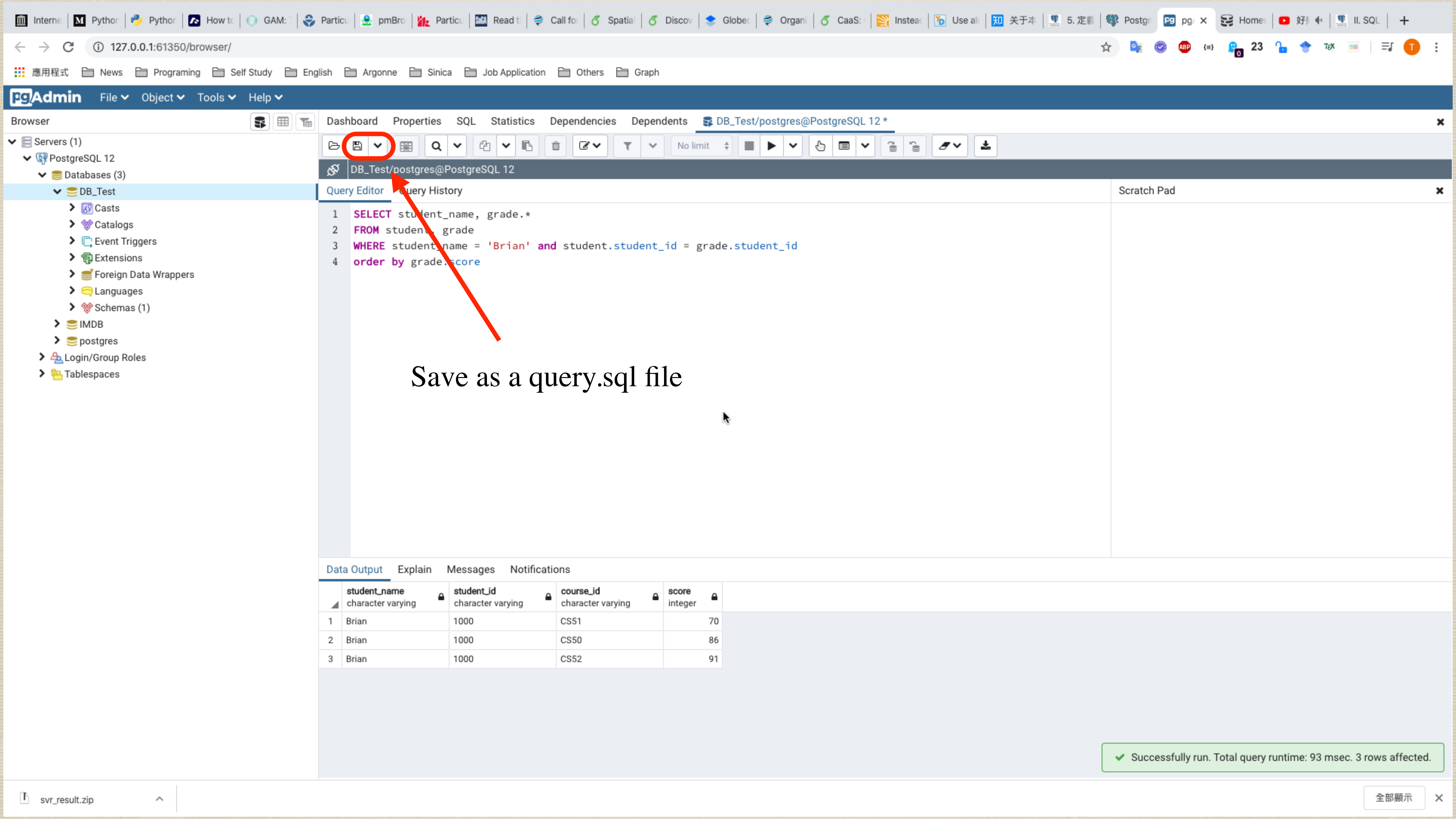
	student_name character varying	student_id character varying	course_id character varying	score integer
1	Brian	1000	CS51	70
2	Brian	1000	CS50	86
3	Brian	1000	CS52	91

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

Query & Result

Ref: SQL Langauge

8. Save your Query (for example)



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

	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

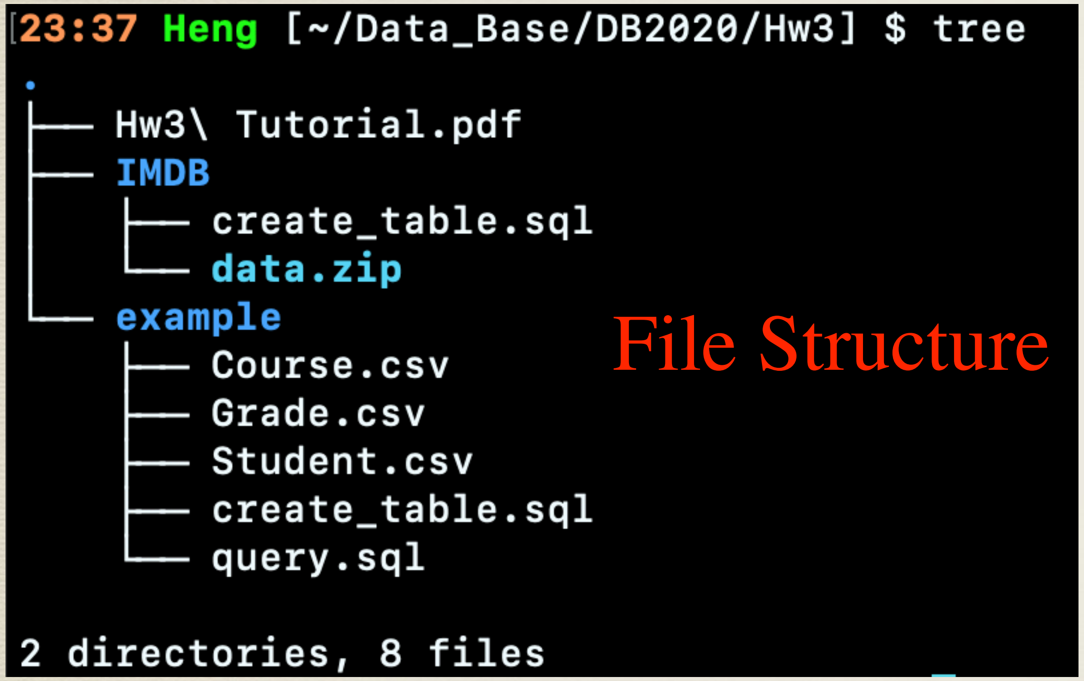
✓ Successfully run. Total query runtime: 93 msec. 3 rows affected.

Save as a query.sql file

Ref: SQL Langauge

9. Submission Rules (for HW)

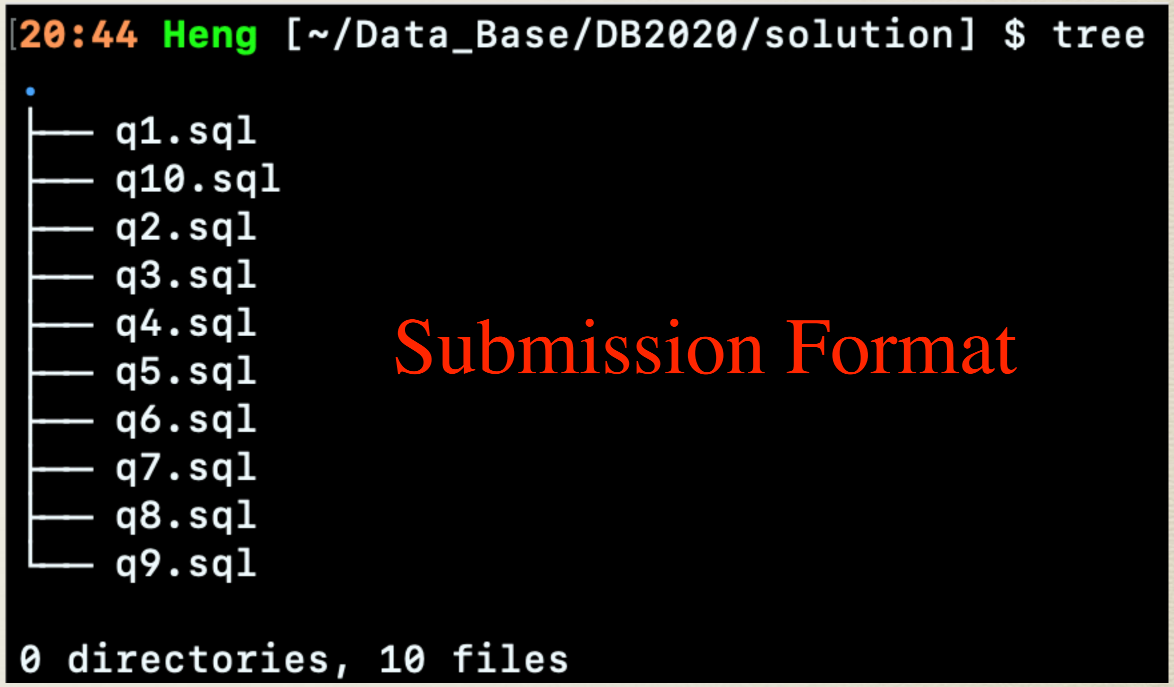
- In your HW package, we have already prepared this document, a template file and a IMDB file for you.



```
[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.