CS 486/586 Introduction to DBMS
Summer 2022

Assignment 0 – Create your database

Due: Please complete by Thursday 06/23. You will need this assignment completed before you watch Videos 2.

Turn in: You do not need to turn in this assignment.

Notes

- Accessing the database through psql is a two-step process.
    a. First, ssh into linux.cs.pdx.edu using your usual MCECS username & password.
    b. Then, at the command line, type:

        psql -h dbclass.cs.pdx.edu -U your_user_name your_user_name

        - Use the db password I give you as the password
        - **Note**: your_user_name is repeated - the first instance is to indicate your username, the second is actually specifying the name of the database to connect to - your db name is the same as your username

- The files from the Google Drive folder should go in the same directory as the directory in which you are running psql. Be sure to use \copy and not copy.
    - Note: copy tries to look for the files on the database machine, which you don't have access to, \copy looks for the files in the directory in which psql is running

Assignment
For this initial assignment, you will create a database. Your database will be similar to the database we use in class. The purpose of this database is so that you can experiment with queries we discuss in class.

1. Access your database: First things first, you'll need to access your database. You can access your database by ssh-ing into linux.cs.pdx.edu and then running the psql command given in the Canvas message. Or you can access the database via the web interface at: https://dbclass.cs.pdx.edu/. Additional information about accessing your database is here: 486/586 Database Access Info. When logging into the database, use the username/password that was given to you on Canvas.

2. Create your database: To create your database, run the series of commands below. You may not understand all these commands quite yet, we'll learn about them throughout the course, but this will get you a database that you can start experimenting with.

    a. First - create your tables and add keys. Note that I didn't add foreign keys so you can play around with outer joins, when we get to them.

    CREATE TABLE books (id int, title text, pagecount int, genre text, authorid int, pubid int);

    ALTER TABLE books add primary key (id);

    CREATE TABLE authors(id int, name text, age int);

    ALTER TABLE authors add primary key (id);

    CREATE TABLE publishers(id int, name text);

    ALTER TABLE publishers add primary key (id);

    CREATE TABLE editions(bookid int, date date, edition text, lang text, pubid int);

    ALTER TABLE editions add primary key(bookid, date);

    For psql: Type in the above command

For Graphical User Interface (GUI): Execute your commands in the SQL command:

b. Second - load your data. The csv files containing the data can be found in the class Google Drive folder here.

For psql: Put the csv files in the same directory where your psql program is running. If you are on linux.cs.pdx.edu, you can use the following commands to get the files into your directory:

you can download to your computer and use scp to transfer the files.

Such as:
scp urlocal/directory/*csv
urUsername@linux.cs.pdx.edu:ur/linux/direcory/

Then run the following commands. Be sure to use \copy, not copy.

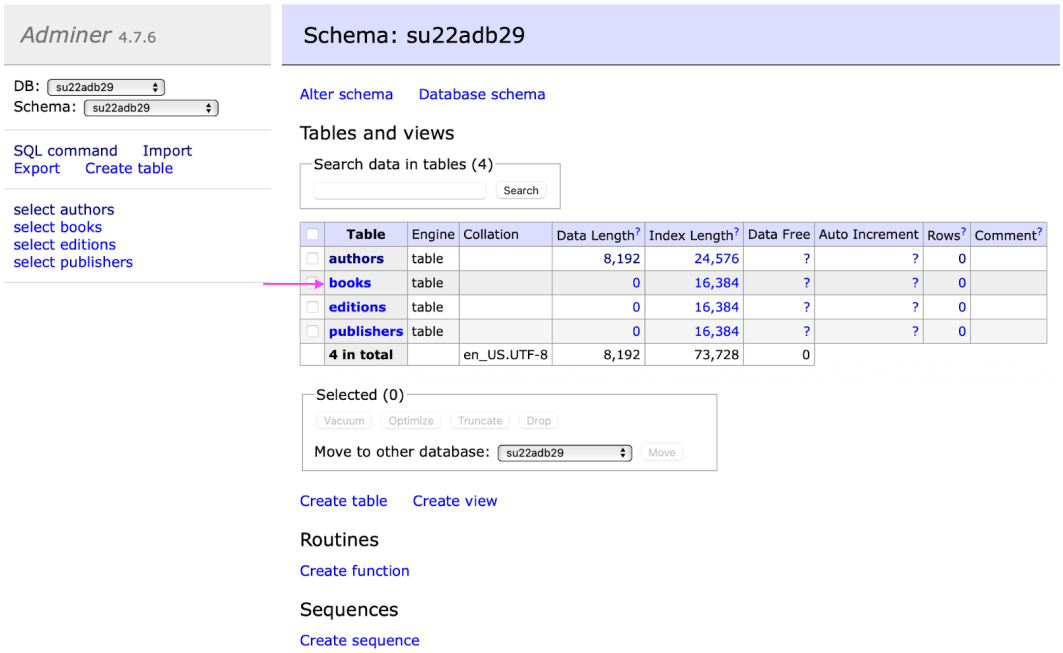
\copy books from books.csv with csv header

\copy authors from authors.csv with csv header

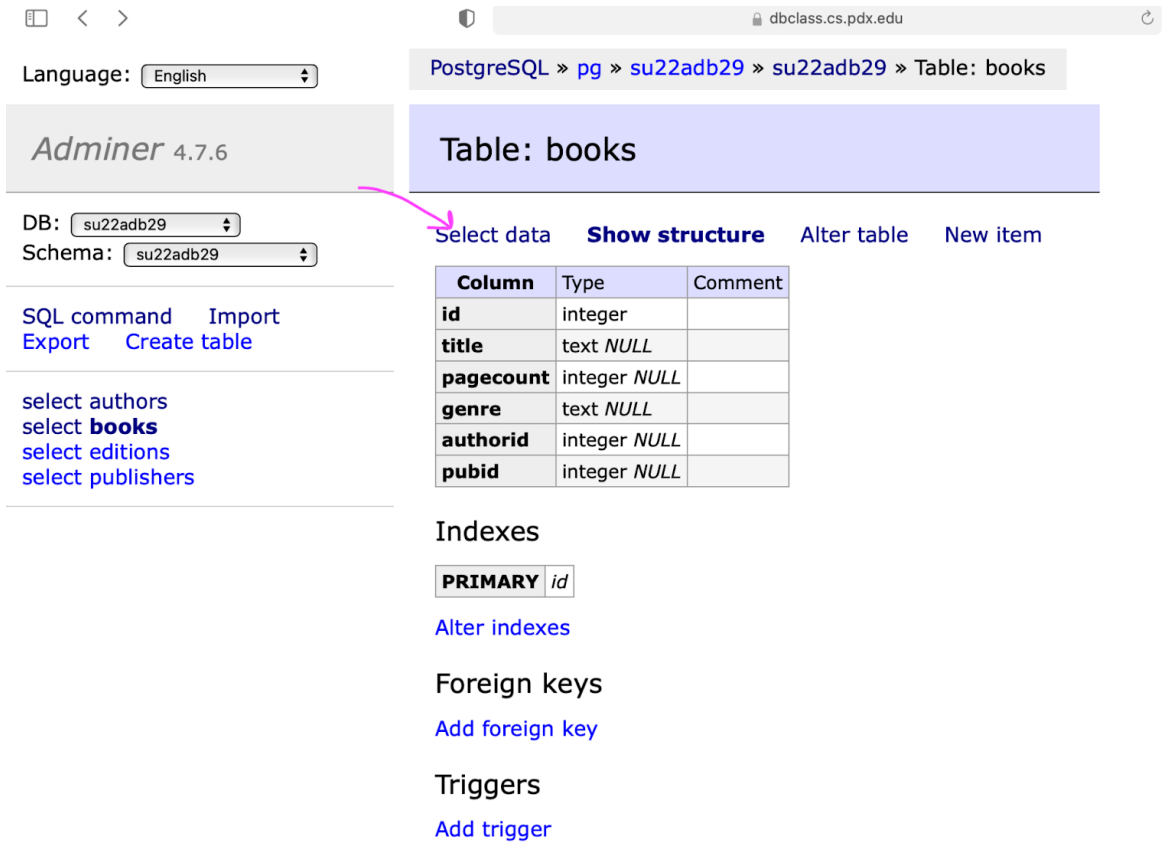
\copy publishers from publishers.csv with csv header

\copy editions from editions.csv with csv header

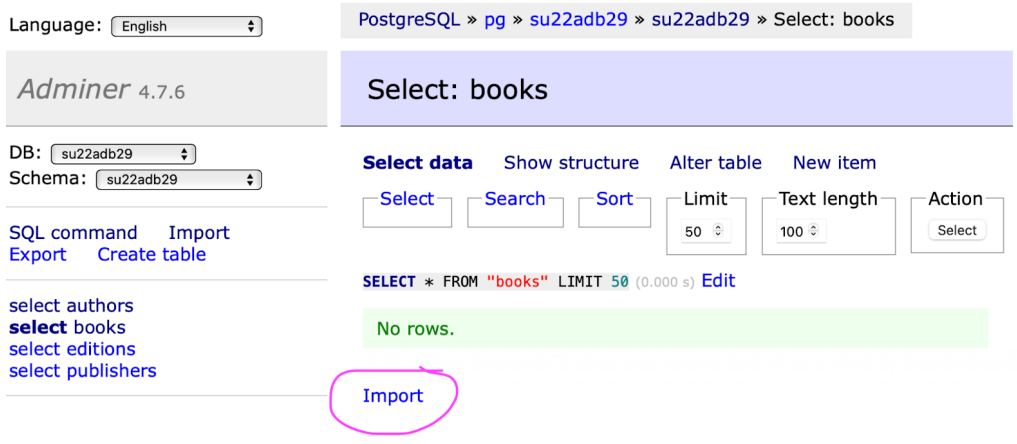
For GUI: Download the four csv files locally and then on Graphical User Interface (GUI) go to Schemas, su22adb_ _ schema that is your username and click on the table you want to add the data to. For example here the table we want to import data in is books.



After clicking on the the table, click on “Select data” on the top left.



And then click on Import as the following example:



This takes you to the page where you can upload your CSV file from your local machine. Choose the relevant CSV file and click import.

Notes:

- See below for some useful PSQL commands, including how to change your password.
- The commands below are for the command line psql program. You may also do this assignment in pgadmin if you wish.
- If you are using pgadmin
    - Use 'SQL command' to type in the queries, tables will appear in the su22adb_ _ schema

Schema Notes:

1. One author can have multiple books
2. One publisher could have published multiple different books
3. Multiple authors could have books published by the same publisher
4. If the author is no more, then their age value is left blank
5. If the book is self published, then the publisher id is 99

Redoing the Tables:

- If you need to reload the data in a table, for example the books table, do:
    DELETE FROM books;
    then use the \copy command above to reload the data

- If you need to totally recreate a table, do:
    DROP TABLE books;
    then use the CREATE command above to recreate the table
    then use the \copy command above to reload the data

Useful PSQL Commands:

\dt - list all tables

\d schemaname.tablename - display info (schema) of a table

\? - help with 'backslash' commands

\q - quit psql

help - help with SQL commands (also can use help select for help on select)

Quotes:

In the syntax Type = "checking" AND Balance > 1000, why are there quotes around checking, but not around 1000 (or Type or Balance).

Type and Balance are attribute names, so do not need quotes. 1000 is an integer or numeric value and doesn't need quotes. But checking is a string, and strings do need quotes. The other error in that syntax is that postgres uses single quotes, rather than double quotes use single quotes in postgres, not double quotes.

Temp Tables:

Postgres allows you to create temporary tables from other tables if necessary – it may be of some use in sanity checking some of the SQL queries you write for your homework assignments.

As an example of how this is done, the syntax:

SELECT first, last INTO agentname FROM agent;

will create a permanent table that only contains the first and last attributes from the agent table. This table will need to be deleted when you are done with it, otherwise it will persist indefinitely. If you want to create a temp table instead, which will be deleted automatically by Postgres when you end your session, you can use the following syntax:

CREATE TEMP TABLE agentname AS SELECT first, last FROM agent;

Change Password:

ALTER USER username WITH PASSWORD ‘newpasswordhere’;