

Practicing SQL

Definitions

Database

A collection of tables (among other objects) that store data

Relational Database

A database that allows and stores connections between data tables using foreign keys

SQL

Structured query language, a way of interacting with databases

PostgreSQL

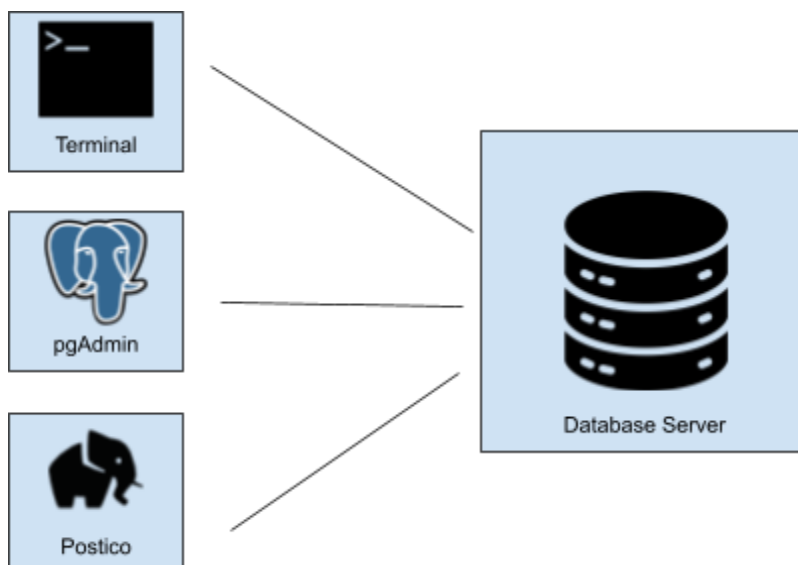
One dialect of SQL that is open source and allows the storage of JSON

Database Server

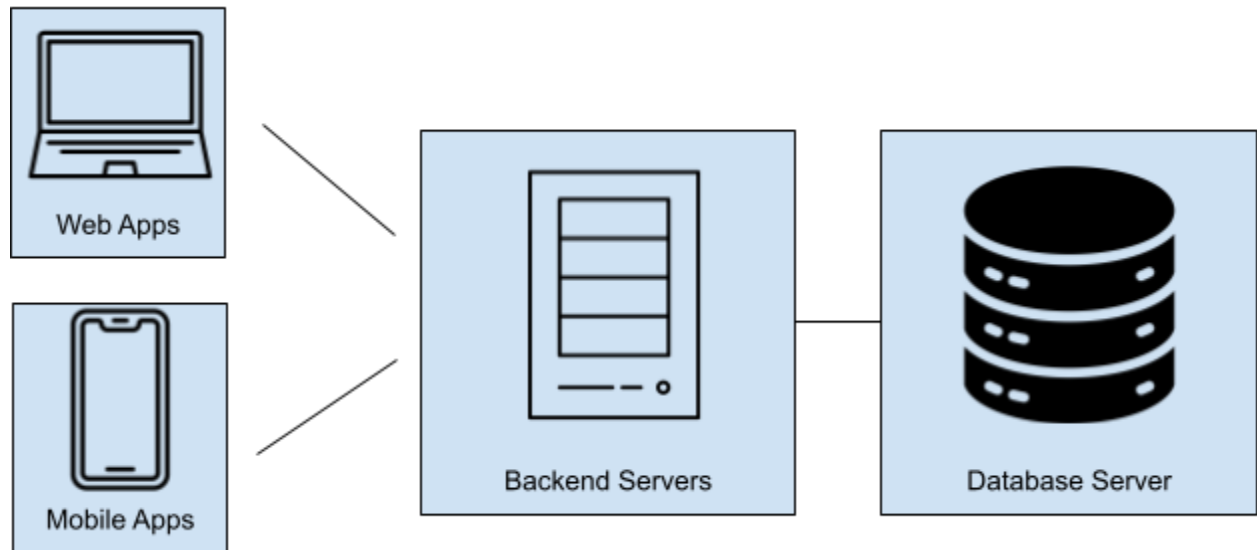
The local or remote server that hosts and allows access to databases

Diagrams

Connecting directly to the database server



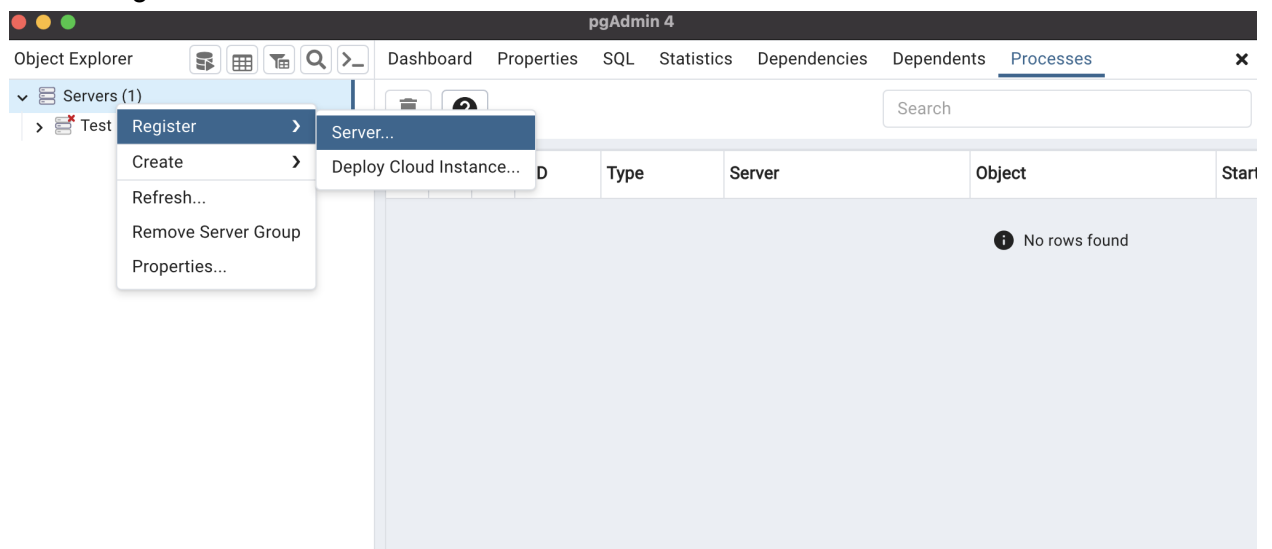
Connecting through an app



Assignment

Connect to Our Test Database

1. Open pgAdmin
2. Right click on Servers in the Object Explorer
3. Select Register -> Server.



4. A dialog will appear. Type a name in the name field of the General tab (anything will work).

Register - Server

General Connection Parameters SSH Tunnel Advanced

Name: Test Server

Server group: Servers

Background: ☐

Foreground: ☐

Connect now?: ☒

Comments:

❗ Either Host name or Service must be specified.

Close Reset Save

5. Select the Connection tab.
6. Enter the host, port (5432), username, database name, and password.

Register - Server

General Connection Parameters SSH Tunnel Advanced

Host name/address: queenie.db.elephantsql.com

Port: 5432

Maintenance database: snyltnzr

Username: snyltnzr

Kerberos authentication?: ☐

Password:

Save password?: ☐

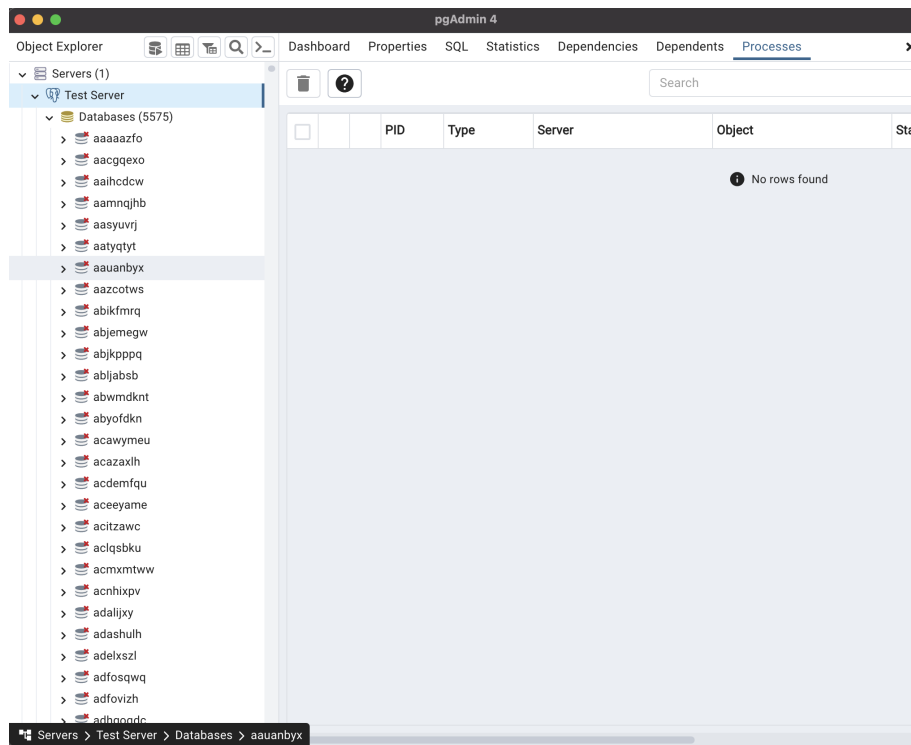
Role:

Service:

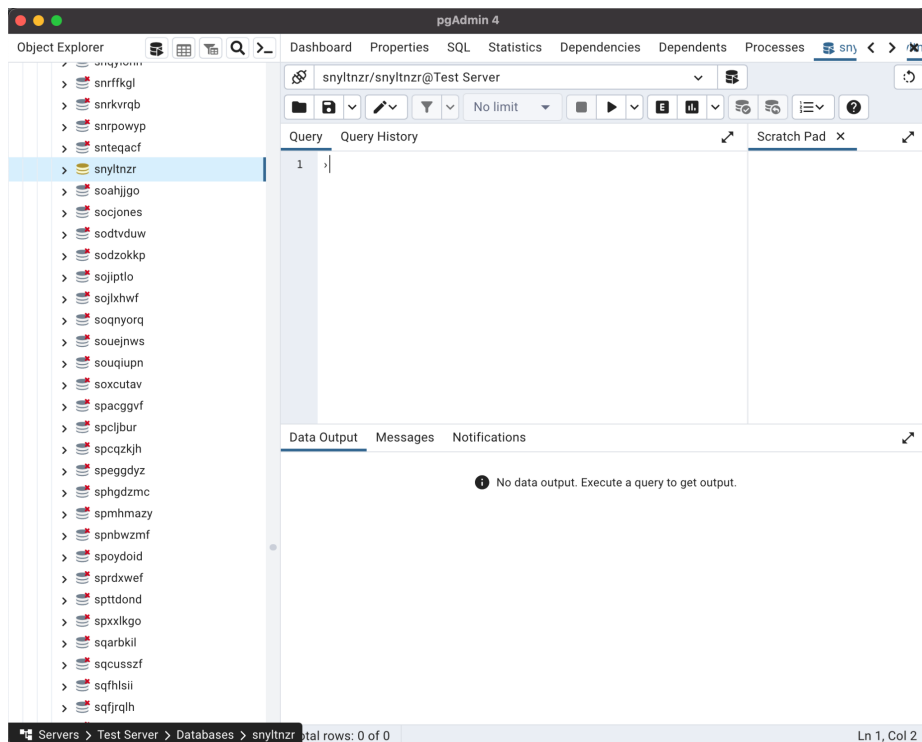
Close Reset Save

7. Click Save.
8. You'll see all of the shared databases in the ElephantSQL server. We only have access to one, though. This is just how pgAdmin works. It shows you all of the databases on the

server. You can find our database by scrolling. We're snyltnzr.



9. Once you find ours, select the query tool icon (the first icon, shaped like a cylinder with a cursor over it). This should open the SQL editor. Now you can play with our database.



Assignment

1. Review SELECT by selecting all rows from the users table.
2. Review SELECT by selecting all addresses from the addresses table.
3. Review UPDATE by changing the email address of a user.
4. Review INSERT by adding yourself as a user.
5. Review INSERT by adding your address. Be sure to include your user_id.
6. Review constraints by adding a WHERE clause to your SELECT statements. Find either yourself or me.
7. Add an ORDER BY clause to SELECT the users and list them alphabetically.
8. Review the JOIN by SELECTing all of the users and their addresses.
9. Create a new table or scores with a primary key called id.
10. Add a column that contains a score value.
11. Add a foreign key to tie each score to a user.
12. INSERT several scores for each user. (This creates a one-to-many relationship because each user can have many scores.)
13. JOIN the data to return the users and their scores.
14. Find the AVG score (the average of all scores in the table).
15. BONUS: Find the AVG score for each user (we didn't go over this).
16. BONUS: Find the highest score in the database using MAX (we didn't go over this).
17. BONUS: Find the highest score for any user in Houston.