

# Web Based Dashboards

## Module # 4 (Deploy Postgre SQL Database)

Dr. Wajahat Gilani

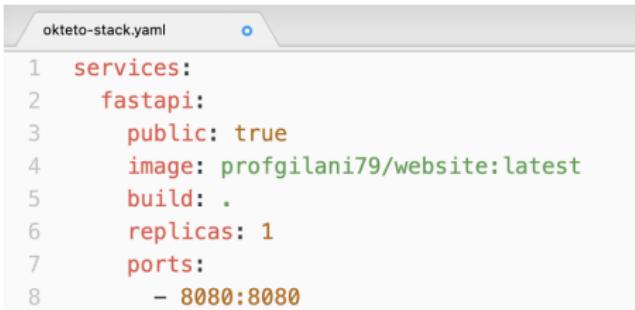
Rutgers Business School

July 1, 2021

## Step # 1: Add Database Information: okteto-stack.yaml

In our application we need a storage for our data, which is why we will now deploy a `postgresql` database under our `fastapi` service. To do that we need to:

- Open Github Desktop
- Add the postgresql information to our `okteto-stack.yaml` configuration file
- Commit the `okteto-stack.yaml` updates to Github Desktop
- Redeploy our project



```
okteto-stack.yaml
1 services:
2   fastapi:
3     public: true
4     image: profgilani79/website:latest
5     build: .
6     replicas: 1
7     ports:
8       - 8080:8080
```

On the left is our original file, below it we will add the additional lines of code that pertains to the `postgresql` database. On a side note, **Okteto** has several different databases that the users can use. **Postgresql** is a popular open-source database, so we will use it.

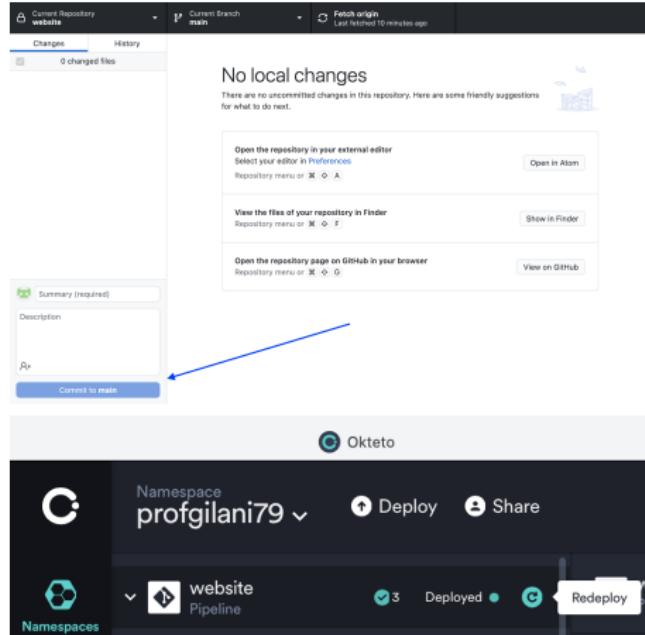
# Step # 1: Add Database Information: okteto-stack.yaml (Continued)

```
okteto-stack.yaml
1 services:
2   fastapi:
3     public: true
4     image: profgilani79/website:latest
5     build: .
6     replicas: 1
7     ports:
8       - 8080:8080
9     environment:
10      - DB_HOST=postgres://pguser:pgpass@postgres:5432/pgdb
11      - secret=dev
12   postgresql:
13     image: bitnami/postgresql:latest
14     ports:
15       - 5432
16     environment:
17       - POSTGRES_USER=pguser
18       - POSTGRES_PASSWORD=pgpass
19       - POSTGRES_DB=pgdb
20     volumes:
21       - data:/bitnami/postgresql
22   volumes:
23     data:
```

(Updated file in canvas)

- Under **fastapi** we add another parameter, **environment**. That is where the **url connection** for the database is defined. It has 4 main parts:
  - ▶ user name (**pguser**)
  - ▶ password (**pgpass**)
  - ▶ database (**pgdb**)
  - ▶ port number (**5432**)
- Those 4 parts are defined in a new entry under **services**, named **postgresql**. Under **postgresql** is the **environment**, that defines the db name, username and password.
- The **port number** 5432 is designated by Okteto, for postgresql databases.

## Step # 2 and #3: Commit Changes and Redeploy



Commit the changes to the `okteto-stack.yaml` file, using [Github Desktop](#).

Now on your [Okteto](#) page, redeploy your project. For example, my application name is [website](#). The curved arrow next to [Deployed](#), will allow me to [Redeploy](#) the project.

# Postgresql Database is Deployed

The screenshot shows the Okteto dashboard interface. On the left sidebar, there are buttons for 'Namespaces' (selected), 'Settings', 'Start Pro Free Trial!', 'Help', and a user icon. The main content area displays a deployment named 'fastapi' within a 'website' pipeline. It lists three pods: 'fastapi' (Running), 'postgresql' (Running), and 'data' (In-use). A 'Logs' tab is selected, showing deployment logs:

```
#10 3.396 Collecting uvicorn==0.14.0
#10 3.410 Downloading uvicorn-0.14.0-py3-none-any.whl (50 kB)
#10 3.736 Installing collected packages: typing-extensions, starlette, pydantic, h11, click, asgiref, uvicorn, fastapi
#10 4.684 Successfully installed asgiref-3.3.4 click-8.0.1 fastapi-0.65.1 h11-0.12.0 pydantic-1.8.2 starlette-0.14.2 typing-extensions-3.10.0 uvicorn-0.14.0
#10 4.684 WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
#10 DONE 5.2s

#11 [7/7] COPY ./app /app
#11 sha256:39f6fd49f8a215849965176b8a03a80efde36cec26095d3ecd26aca7703f2e3
#11 DONE 0.3s

#12 exporting to image
#12 sha256:e8c613e07b0b7ff33893b694f7759a10d42e180f2b4dc349fb57dc6b71dcab00
#12 exporting layers
```

On the right, a summary shows 2/10 pods, 3GB/5GB storage, and a 'Deployed' status.

Your postgresql database is now deployed and ready to use within your website application.

# Connect to database

```
wajgilani@1053macbooks-MacBook-Pro website % kubectl port-forward service/postgresql 5432:5432  
Forwarding from 127.0.0.1:5432 -> 5432  
Forwarding from [::1]:5432 -> 5432
```



Your postgresql database is now deployed and ready to use within your website application.

# Download Postgresql GUI



## DBeaver Community

Free Universal Database Tool

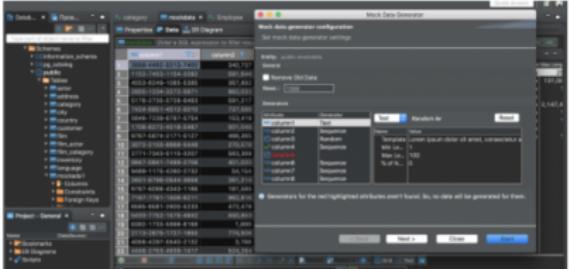
Star 21,109 Follow @dbeaver\_news

search here ... Go

Home About Download Sources Documentation News Support Enterprise Edition CloudBeaver

## Universal Database Tool

Free multi-platform database tool for developers, database administrators, analysts and all people who need to work with databases. Supports all popular databases: MySQL, PostgreSQL, SQLite, Oracle, DB2, SQL Server, Sybase, MS Access, Teradata, Firebird, Apache Hive, Phoenix, Presto, etc.



<https://dbeaver.io>

# Download Postgresql GUI

## Download

### Community Edition 21.1.1

Released on June 21, 2021 ([Milestones](#)).

It is free and open source ([license](#)).

Also you can get it from the [GitHub mirror](#).

### Enterprise Edition 21.1

Released on June 7, 2020

EE version web site: [dbeaver.com](#)

Trial version is available.

#### Windows

- [Windows 64 bit \(installer\)](#)
- [Windows 64 bit \(zip\)](#)
- [Install from Microsoft Store](#)
- [Chocolatey \(choco install dbeaver\)](#)

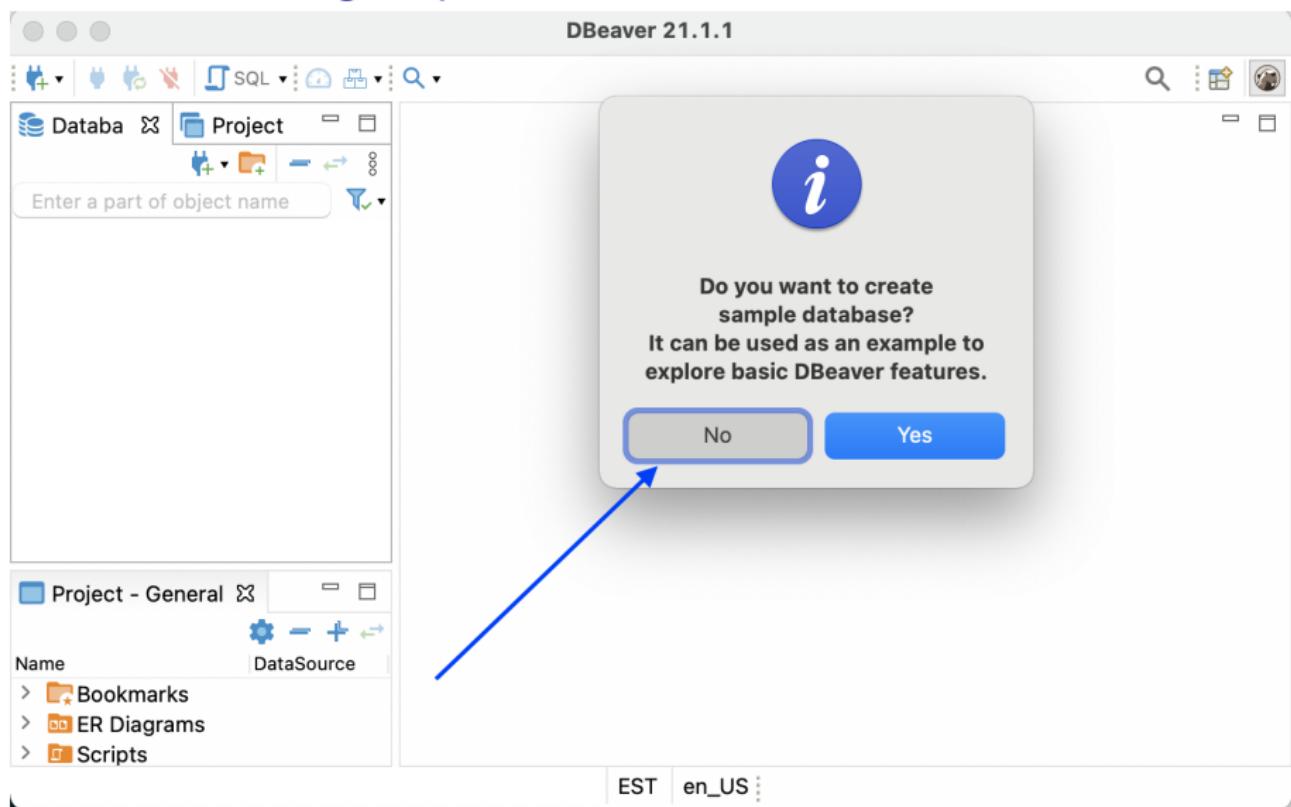
#### Mac OS X

- [Mac OS X \(dmg\)](#)
- [Mac OS X \(zip\)](#)
- [Brew Cask \(brew install --cask dbeaver-community\)](#)
- [MacPorts \(sudo port install dbeaver-community\)](#)

#### Enterprise Edition features:

- Support of NoSQL databases:
  - MongoDB
  - Cassandra
  - InfluxDB
  - Redis
  - Amazon DynamoDB
  - Amazon DocumentDB
  - Amazon Keyspaces
  - Google Bigtable
  - Couchbase
  - CouchDB
- Advanced extensions for:
  - Oracle
  - SQL Server
  - Netezza

# Download Postgresql GUI



Open and click no about creating sample database

# Download Postgresql GUI

Connect to a database

Select your database

Create new database connection. Find your database driver in the list below.

Type part of database/driver name to filter

Sort by:  Title  Score

Popular

- SQL
- NoSQL
- Analytical
- Timeseries
- Embedded
- Hadoop / BigData
- Full-text search
- Graph databases

DB2 LUW    DuckDB    MariaDB    MySQL    ORACLE    PostgreSQL    Microsoft SQL Server

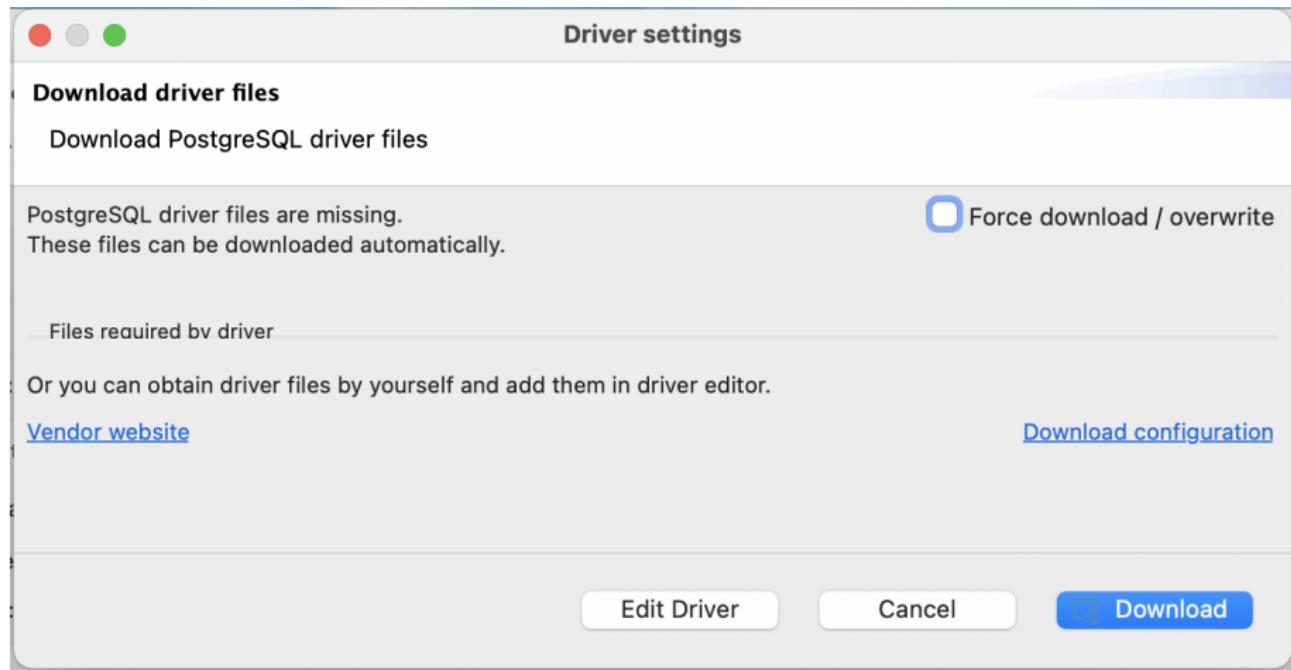
SQLite    Apache Drill    Apache Hive    Apache Ignite    Apache Phoenix

Test Connection ...    < Back    Next >    Cancel    Finish

A screenshot of a Mac OS X-style application window titled "Connect to a database". The window has a sidebar on the left with a "Popular" section containing links for SQL, NoSQL, Analytical, Timeseries, Embedded, Hadoop / BigData, Full-text search, and Graph databases. The main area is titled "Select your database" and contains a search bar and a sorting dropdown. Below the sidebar is a grid of database icons. A blue arrow points upwards from the bottom right towards the PostgreSQL icon. The icons include: DB2 LUW (green square with white text), DuckDB (yellow circle with black dot), MariaDB (lizard logo), MySQL (blue logo with cursor), Oracle (red logo), PostgreSQL (blue logo with red flame), Microsoft SQL Server (red logo with yellow flame), SQLite (blue feather icon), Apache Drill (drill bit icon), Apache Hive (bee icon), Apache Ignite (red flame icon), and Apache Phoenix (orange flame icon). At the bottom are buttons for "Test Connection ...", navigation arrows, and "Cancel" and "Finish" buttons.

Select the type of database we are connecting to.

# Download Postgresql GUI



Download the driver settings.

# Download Postgresql GUI

Connect to a database

Connection Settings  
PostgreSQL connection settings

Main PostgreSQL Driver properties SSH Proxy SSL

Server

Host: localhost Port: 5432

Database: pgdb

Authentication

Authentication: Database Native

Username: pguser

Password:  Save password locally

Advanced

User role: Local Client: <not present>

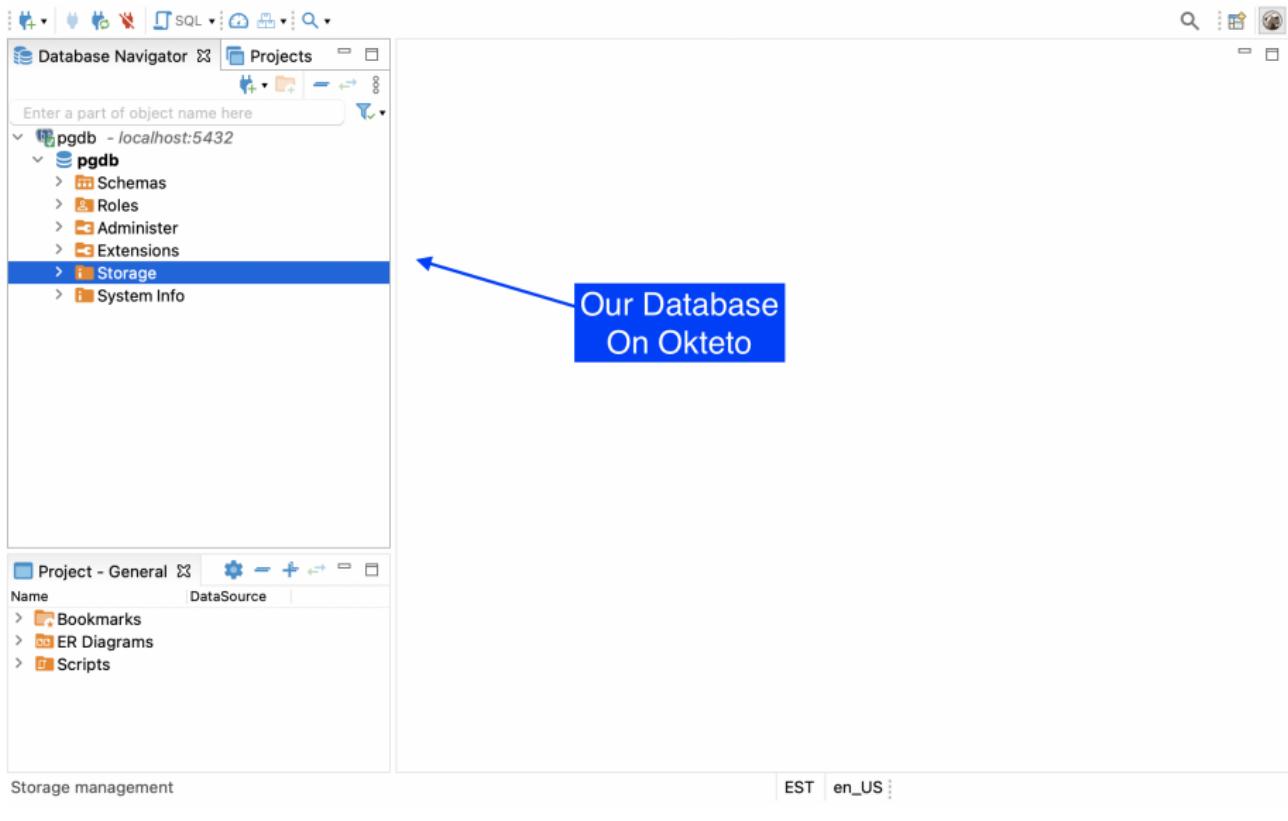
① You can use variables in connection parameters. Connection details (name, type, ... )

Driver name: PostgreSQL Edit Driver Settings

Test Connection ... < Back Next > Cancel Finish

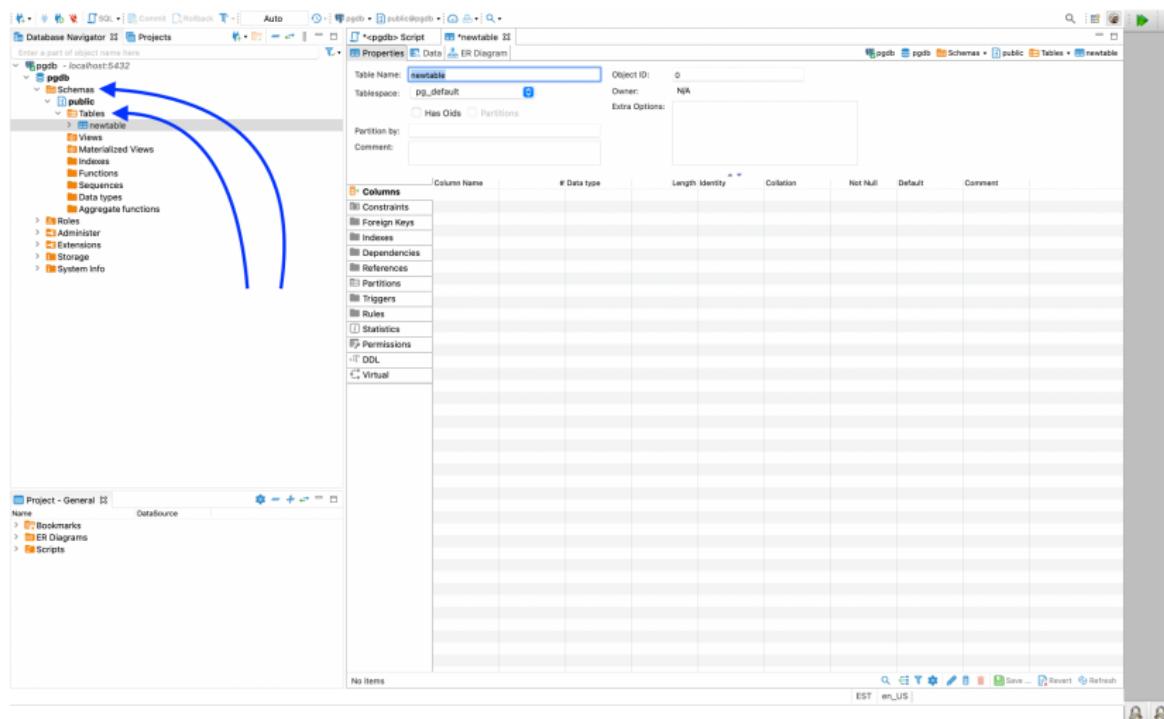
Dr. Wajahat Gilani (Rutgers Business School) Web Based Dashboards July 1, 2021 12 / 19

# Download Postgresql GUI



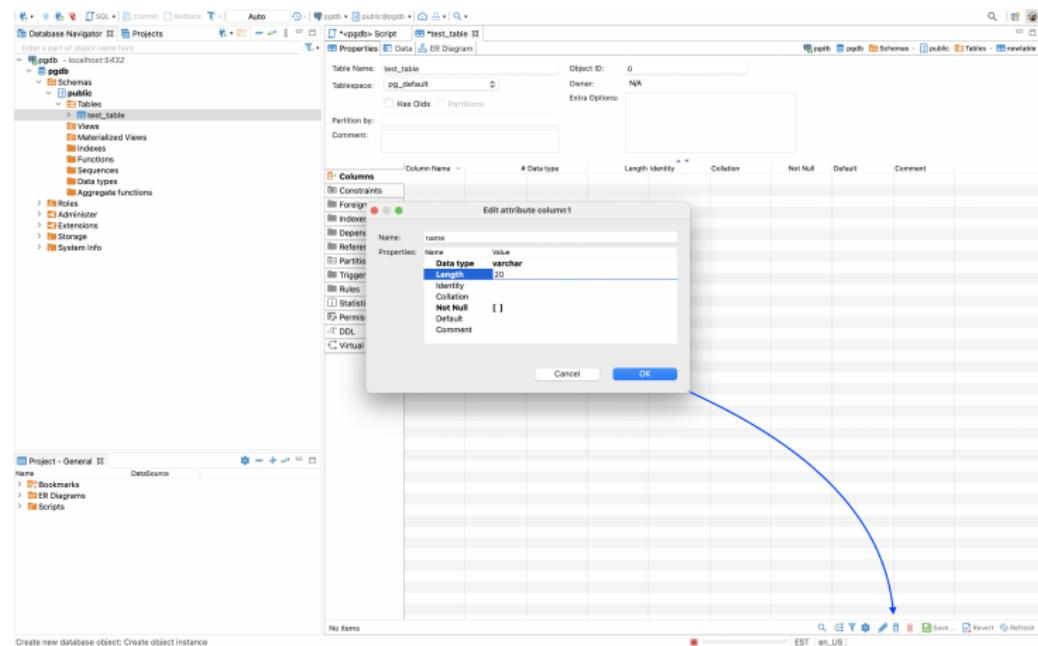
We are connected to our database in Okteto!!!!

# Create Example Table - test\_table



Click on **Schemas**, and then right-click on **Tables**. A form opens up that will allow us to name our table and create columns within that table.

# Create Example Table - test\_table



Change the name to **test\_table**, then click on the bottom button to add columns. When creating columns in databases, you must specify the **data type**. Varchar is a column for alpha-numerics, and we specified that it will be up to 20 characters.

# Create Example Table - test\_table

The screenshot shows the pgAdmin 4 interface for creating a table named 'test\_table'. The table has one column named 'age' with a data type of 'varchar(20)'. A tooltip 'Edit attribute column1' is displayed, showing the properties for the 'age' column, including the data type dropdown which has 'int' selected. The pgAdmin interface includes a sidebar with various database objects like schemas, tables, and functions.

Create the column **age**, that will be an **int**, click on the arrow next to the choices and choose it.

Create the

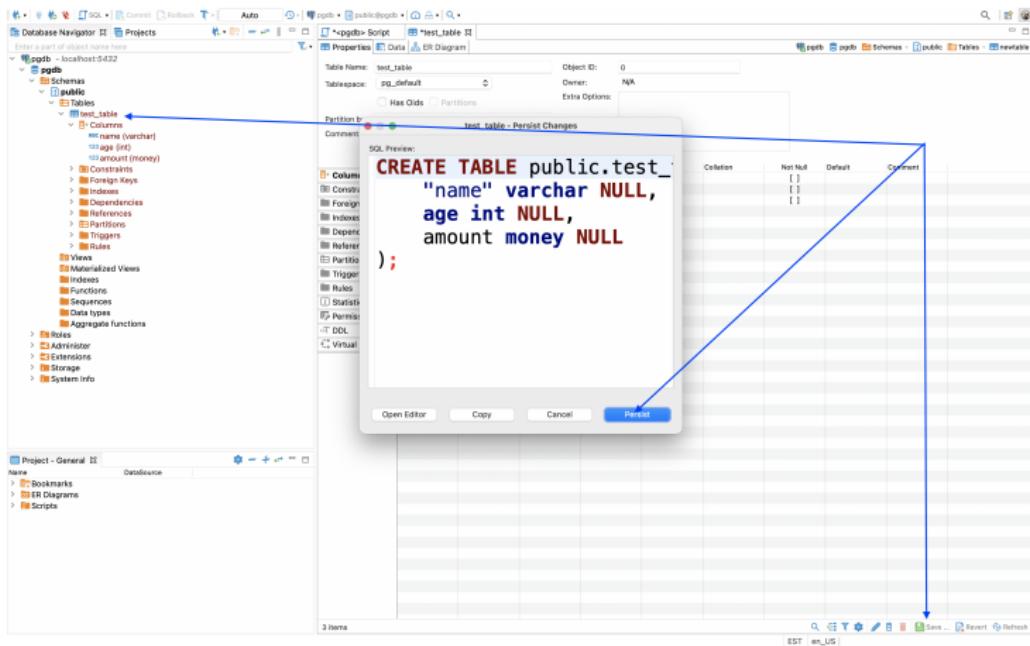
# Create Example Table - test\_table

The screenshot shows the pgAdmin 4 interface for creating a new table. The left sidebar shows the database structure under 'pgdb'. The main window shows the 'Properties' tab for a table named 'test\_table' in the 'public' schema. A tooltip is displayed over the 'amount' column, showing its properties: Name: amount, Data type: money.

Column Name	Data type	Length	Identity	Default	Comment
int_name	varchar	20		1	
amount	money				

Same with **amount**, it will be the datatype **money**.

# Create Example Table - test\_table



Save the table and click on persist. The table has now been created. Now lets load up some data.

# Create Example Table - test\_table

The screenshot shows the pgAdmin 4 interface. On the left, the Database Navigator displays a project named 'pgAdmin - localhost:5432' containing a schema 'public' with a table 'test\_table'. The table has three columns: name (text), age (int), and amount (money). A 'Script' tab contains the following SQL code:

```
select * from test_table
insert into test_table values('batman',30,1000000);
insert into test_table values('superman',29,5000);
insert into test_table values('wonderwoman',300,13);
```

On the right, the 'Query' tab shows the results of the 'select \* from test\_table' query, displaying three rows of data:

	name	age	amount
1.	Batman	30	\$1,000,000.00
2.	superman	29	\$5,000.00
3.	wonderwoman	300	\$13.00

```
insert into test_table values('batman',30,1000000);
insert into test_table values('superman',29,5000);
insert into test_table values('wonderwoman',300,13);
```

Highlight the SQL commands, and then press **control + enter** to run the **insert statements**. Then run the **select** command and you should see all the results.

Now lets load up some test data, just so we can do a select command. Write the following SQL insert commands:

# Close Database and Connection

When you are done, just close DBeaver, and then press **control + C**, to exit out of Okteto's connection to the database.

```
wajgilani@1053macbooks-MacBook-Pro website % kubectl port-forward service/postgresql 5432:5432
Forwarding from 127.0.0.1:5432 -> 5432
Forwarding from [::1]:5432 -> 5432
Handling connection for 5432
^C%
wajgilani@1053macbooks-MacBook-Pro website %
```