Database 2
Midterm
Feb. 06, 2020

Prerequisites:

- This activity assumes a prior knowledge of:
    - Command line usage
    - Basic Javascript
    - Basic Node.js and npm
    - Git and Github
    - PostgreSQL
    - Heroku (Database as a Service)

Goals:

- This walkthrough will have three parts:
    - Setting up Github repository
    - Setting up remote PostgreSQL database using Heroku
    - Setting up a Node.js environment to integrate PostgreSQL remote database

NOTE: **To learn more about Node.js and PostgreSQL integration see documentation**
https://node-postgres.com/

Part 1: Setting up **Github** repository

- Create new github repo “nodepg”
- Create README.md and use this as your documentation file where you can put your learnings and other important notes here.
  Example:

~~~
## This is my activity in database 2 and the important codes/snippets are the following:
```js
console.log(“asdfa”)
```

## In case of error [Type of error], here are the steps that you need to consider:
1. Step 1
2. Step 2
~~~

Part 2: Setting up remote PostgreSQL database using Heroku

- Login to your heroku and list down all your postgreSQL database credentials
  Host: ****
  Database: ***
  User: ****
  Port: ****
  Password: ***
- Launch PGAdmin4 and create the following sample table and insert the given record

```
CREATE TABLE books (
ID SERIAL PRIMARY KEY,
author VARCHAR(255) NOT NULL,
title VARCHAR(255) NOT NULL );

INSERT INTO books (author, title) VALUES ('J.K. Rowling', 'Harry Potter');
```

- Create a .env file and write the following data:

```
DB_USER=api_user
DB_PASSWORD=password
DB_HOST=localhost
DB_PORT=5432
DB_DATABASE=books_api
```

Part 3: Setting up a Node.js environment to integrate PostgreSQL remote database

- Initialize the repo as npm environment using the command `npm init -y`
- Install the PostgreSQL dependencies `npm install --save pg`
- Create new file index.js

```
const { Pool } = require('pg')

const pool = new Pool({
    user: `${process.env.DB_USER}`,
    host: `${process.env.DB_HOST}`,,
    database: `${process.env.DB_DATABASE}`,
    password: `${process.env.DB_PASSWORD}`,
    port: `${process.env.DB_PORT}`,
    ssl: true,
  })

pool.query('SELECT * FROM books', (error, results) => {
    if (error) {
        throw error
    }
    console.log(results.rows)
})
```

- Execute the script `npm run test-nodepg`

  Terminal Output:

```
[ { id: 1, author: 'J.K. Rowling', title: 'Harry Potter' } ]
```

Challenge:

Consider our previous activity (see link). Solve the problems stated in the reference site and provide the required scripts.

1. `npm run lastname-prob1` :Get all pet_name's that are owned by owners with the occupation of "Programmer"
2. `npm run lastname-prob2` :Get The Occupation, name, and class fields for all rows where the pet is a Bird.
3. `npm run lastname-prob3` :Get the all the rows where the owner is a female OR the class is a Reptile. ONLY when the pet has an owner
4. `npm run lastname-prob4` :Same as 3, but including pets that are Reptiles without homes.
5. `npm run lastname-prob5` :Get all rows where the owner is a male, even if he does not currently own a pet.
6. `npm run lastname-prob6` :Get all owners that do not own a pet
7. `npm run lastname-prob7` :Get all pets that do not have an owner

Link: https://truthseekers.io/lessons/sql-one-to-many-relationships-and-joins/