Projekt: System Zarządzania Partią Polityczną

Bohdan Shcherbak
Instytut Informatyki Uniwersytetu Wrocławskiego
11 czerwca 2019

Ten plik jest dokumentacją do projektu z przedmiotu Bazy Danych. Jest w nim diagram E-R do tej bazy, opis więzów pominiętych w diagramie, opis uprawnień użytkowników init i app, opis sposobu zaimplementowania poszczególnych funkcji API oraz tego jak należy program uruchamiać.

Skrypt jest napisany w języku JavaScript i odpowiada on specyfikacji podanej w tym dokumencie. Zanim przejdziesz dalej, niezbędne jest przeczytanie wspomnianego dokumentu dla zrozumienia niniejszej dokumentacji.

Przed uruchomieniem projektu należy zainstalować Node.js.

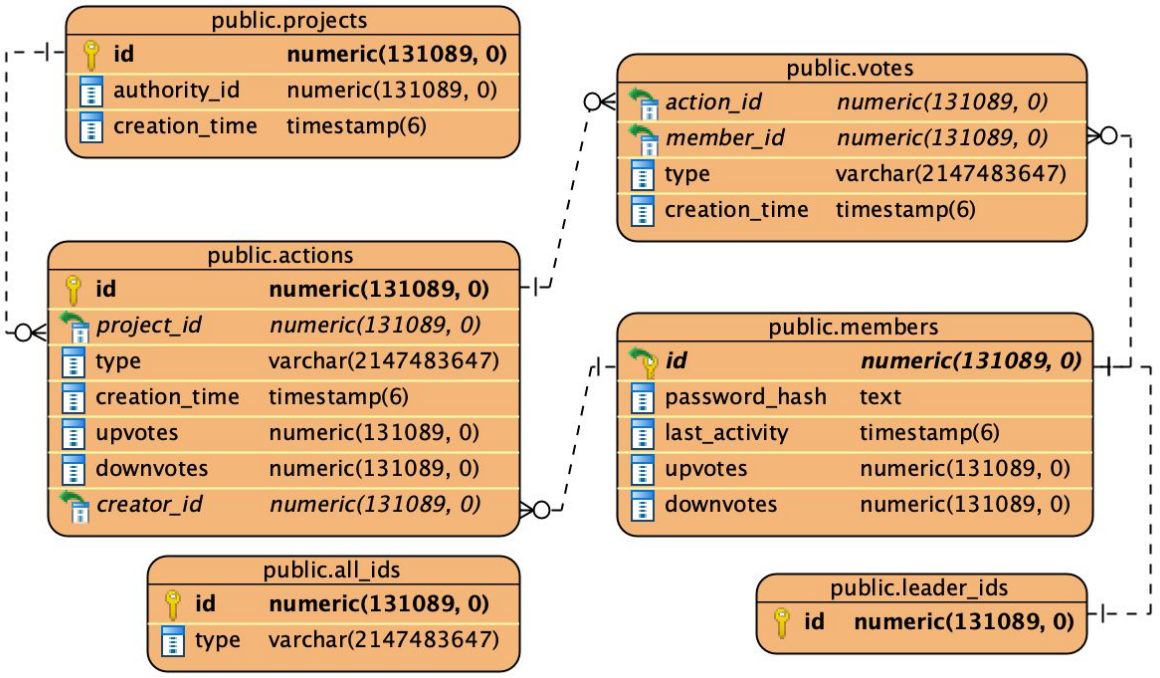
Linki typu [link do specyfikacji] są linkami do elementów na stronie ze specyfikacją.

Przy podaniu na wejście czegokolwiek różniącego się od wywołań API zgodnych ze specyfikacją razem ze znakiem nowego wiersza, na wyjście zostanie wypisany wynik `{"status":"ERROR"}`, w przypadku jeśli na wejście do będzie podane coś, co nie jest poprawnym obiektem JSON, zostanie również wypisany komunikat o błędzie SyntaxError, ale program nie zakończy działania.

Po wywołaniu skryptu z parametrem --init i poprawnym użyciem funkcji API open, wykonuję on sql skrypt physical_model.sql w korzeniu projektu, który stwarza kilka pomocniczych typów enum, tabele, zaprezentowane w poniższym modelu konceptualnym, funkcje pełniące główną rolę w działaniu zaimplementowanych funkcji API, stwarza również użytkownika app, któremu nadaje tylko i wyłącznie te wszystkie prawa niezbędne do wykonania wszystkich tych sql funkcji oprócz funkcji leader. Niepowodzenie w wykonaniu tego skryptu (np. jeśli te tabele są już w bazie danych) powoduje wypisanie komunikatu o błędzie i zakończenie działania programu. PostgreSQL użytkownik init, który musi istnieć przed tym wywołaniem skryptu physical_model.sql, ma mieć dotyczące zaznaczonej w wywołaniu funkcji API open bazy danych prawa stwarzania nowych użytkowników, tabeli, funkcji i typów (enum) i ma mieć prawa nadania praw wykonywania funkcji i wykonywania operacji SELECT, DELETE i UPDATE na wszystkich utworzonych w tej bazie danych tabelach. Do szyfrowania haseł używam modułu pgcrypto, który musi być zainstalowany w bazie danych przed uruchomieniem programu (`CREATE EXTENSION pgcrypto;`).

Model konceptualny

Diagram E-R (stworzony przy pomocy programu Visual Paradigm)



Na powyższym diagramie warto zignorować wszystkie znaczenia podane w nawiasach, a typ varchar(2147483647) przy polach type oznacza typ enum zdefiniowany następująco:

- tabela all_ids: `CREATE TYPE id_type AS ENUM ('member', 'action', 'project', 'authority');`
- tabela votes: `CREATE TYPE vote_type AS ENUM ('up', 'down');`
- tabela actions: `CREATE TYPE action_type AS ENUM ('support', 'protest');`

Linie przerywane wskazują na pola, które są w relacji.

oznacza relację one to one. oznacza relację one to many. oznacza relację many to one.	oznacza klucz. oznacza klucz obcy. oznacza zwykłą kolumnę.

Więzy i zależności

Każda wartość, która się znajduję w dowolnej kolumnie o nazwie id w tych tabelach znajduje się również w kolumnie id (która jest primary key) tabeli all_ids, a w odpowiednim wierszu kolumny type jest “typ” tego id, czyli informacja czy ten id odpowiada w akcji, projektu, głosowi, członkowi czy autorytetowi.
W tabeli leader znajdują się id członków, które są liderami.

Funkcje API

open - wywołanie tej funkcji musi być wyłącznie pierwsze po uruchomieniu programu. Jeśli tego się nie stanie, każde wywołanie dowolnej funkcji API będzie zwracało wynik {"status":"ERROR"}. Próbuję ono nawiązać połączenie z zaznaczoną bazą danych używając podanych danych uwierzytelniających. W razie nieudanej próby połączenia z bazą danych, aplikacja wypiszę {"status":"ERROR"}, po czym każde kolejne wywołanie dowolnej funkcji będzie zwracało {"status":"ERROR"}. [link do specyfikacji]

leader - wywołuję sql funkcję leader(creation_time TIMESTAMP, password TEXT, member_id NUMERIC). [link do specyfikacji]

support - wywołuję sql funkcję support(creation_time TIMESTAMP, member_id NUMERIC, password TEXT, action_id NUMERIC, project_id NUMERIC, authority_id NUMERIC DEFAULT null). [link do specyfikacji]

protest - wywołuję sql funkcję protest(creation_time TIMESTAMP, member_id NUMERIC, password TEXT, action_id NUMERIC, project_id NUMERIC, authority_id NUMERIC DEFAULT null). [link do specyfikacji]

upvote - wywołuję sql funkcję upvote(creation_time TIMESTAMP, member_id NUMERIC, password TEXT, action_id NUMERIC). [link do specyfikacji]

downvote - wywołuję sql funkcję downvote(creation_time TIMESTAMP, member_id NUMERIC, password TEXT, action_id NUMERIC). [link do specyfikacji]

actions - wywołuję sql funkcję get_actions(creation_time TIMESTAMP, member_id NUMERIC, password TEXT, _type action_type DEFAULT null, project_or_authority id_type DEFAULT null, project_or_authority_id NUMERIC DEFAULT null) i wypisuję wynik w postaci JSON obiektu. [link do specyfikacji]

projects - wywołuję sql funkcję get_projects(creation_time TIMESTAMP, member_id NUMERIC, password TEXT, _authority_id NUMERIC DEFAULT null) i wypisuję wynik w postaci JSON obiektu. [link do specyfikacji]

votes - wywołuję sql funkcję get_votes(creation_time TIMESTAMP, _member_id NUMERIC, password TEXT, action_or_project id_type DEFAULT null, action_or_project_id NUMERIC DEFAULT null) i wypisuję wynik w postaci JSON obiektu. [link do specyfikacji]

trolls - wywołuję sql funkcję get_trolls(creation_time TIMESTAMP) i wypisuję wynik w postaci JSON obiektu. [link do specyfikacji]

Uruchamianie programu

Sposób 1:

`node app.js [--init]`

W przypadku jeśli jest przygotowany plik z wywołaniami API w formacie i kolejności odpowiadającej specyfikacji, w terminalach systemu UNIX i w PowerShell systemów Windows te wywołania API można uruchomić w sposób następujący.

Sposób 2:

`cat /path/to/file.txt | node app.js [--init]`

gdzie zakładam, że /path/to/file.txt jest ścieżką do wyżej wspomnianego pliku.

Zakończenie programu

Program przestanie wczytywać znaki z wejścia przy napotkaniu znaku EOF, który, będąc w linii poleceń, można wywołać przy pomocy skrótu klawiszowego Ctrl+D na systemach UNIX albo skrótu Ctrl+Z na systemach Windows. Przy drugim sposobie uruchomienia programu program zakończy działanie automatycznie po wczytaniu całego pliku i wykonaniu wszystkich wywołań funkcji API, w nim zawartych.