

# Projekt: System Zarządzania Partią Polityczną

Bazy Danych 2019

Instytut Informatyki  
Uniwersytet Wrocławski

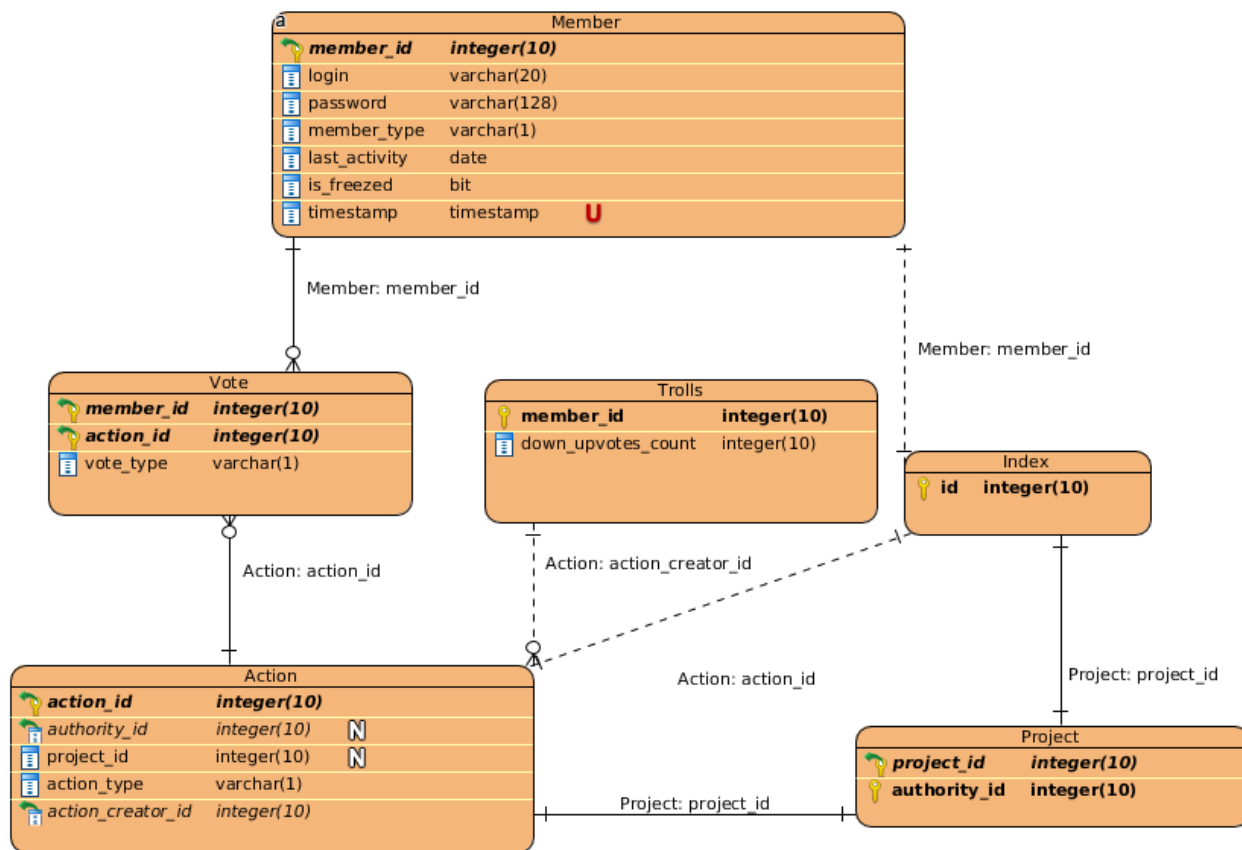
Adam Kufel

## 1. Model konceptualny i fizyczny bazy danych

Baza danych składa się z sześciu tabel przedstawionych na diagramie E-R poniżej. Diagram należy rozumieć jako punkt wyjścia do implementacji bazy danych. Ponadto, implementacja powinna uwzględnić następujące kwestie:

- 1) Członek partii (patrz tabela *Member*) może głosować tylko wtedy, gdy jego konto nie jest zamrożone, tj. wartość pola *is\_active* jest równa 1. Jest ona wyliczana na podstawie różnicy kodów czasowych z argumentu bieżącej akcji, a ostatnim, zapisanym kodem czasowym jego poprzedniej akcji w tabeli *Member*. Jeśli wynik jest mniejszy niż rok, wówczas wartość w tabeli jest zastępowana nowym kodem czasowym. W przeciwnym wypadku konto jest zamrożone i należy zwrócić błąd wykonania działania użytkownika.
- 2) Pole *Member.member\_type* dopuszcza dwie możliwe przyjmowane wartości: 'L' (*leader*), 'M' (*member*).
- 3) Pole *Vote.vote\_type* przyjmuje dwie możliwe wartości: 'Y' (*yes* – głos popierający akcję), 'N' (*no* – głos przeciwko akcji)
- 4) Pole *Action.action\_type* przyjmuje dwie możliwe wartości: 'S' (*support* – akcja wspierająca dane działanie projektu), 'P' (*protest* – akcja przeciwna danemu działaniu w projekcie)
- 5) Akcja może być autoryzowana w dwojaki, rozłączny sposób:
  - bezpośrednio organem władzy (wtedy pole *Action.project\_id* jest NULL)
  - poprzez indentyfikator projektu (wtedy pole *Action.authority\_id* jest NULL)
- 6) Identyfikatory organów władzy *authority\_id* to unikalne i niezmiennie stałe. Dlatego należy sprawdzać, czy nowo podany identyfikator nie powtarza się. W tym celu stworzona została tabela *Index*, w której trzymane są wszystkie identyfikatory posortowane rosnąco. Dzięki temu sprawdzenie można sprawnie wykonać za pomocą wyszukiwania binarnego.
- 7) Tabela *Trolls* przechowuje sumaryczny bilans głosów popierających i przeciwnych dla wszystkich akcji danego użytkownika. Gdy głos zostanie oddany na akcję, wtedy jest odnajdywany jej id twórcy, za pomocą którego można zwiększyć bądź zmniejszyć licznik o 1.

## 2. Diagram E-R



Rys. 1. Diagram E-R Systemu Zarządzania Partią Polityczną

## 3. Opis funkcji API

### 1. Funkcje liderów:

- `votes <timestamp> <member> <password> [ <action> | <project> ]` - zwraca listę wszystkich członków wraz z sumarycznymi liczbami oddanych przez nich głosów za oraz przeciw. W przypadku podania argumentu *action* lub *project* należy ograniczyć się do głosów przynależących do podanej wartości.
- `projects <timestamp> <member> <password> [ <authority> ]` - zwraca listę wszystkich projektów wraz z id organu władzy będącym właścicielem projektu. W przypadku podania argumentu *authority* należy zwrócić projekty przynależące do tego organu władzy.
- `actions <timestamp> <member> <password> [ <type> ] [ <project> | <authority> ]` - zwraca listę wszystkich akcji w zależności od podanych argumentów. Argumenty *member* oraz *password* służą do autoryzacji lidera. W sytuacji, gdy podano *project* bądź *authority*, należy zwrócić akcje przynależące do jednego bądź drugiego. Tak samo w przypadku argumentu *type* należy zwrócić akcje przynależące do danego typu

## 2. Funkcje użytkownika:

- `support/protest <timestamp> <member> <password> <action> <project> [ <authority> ]` - skutkuje dodaniem nowej akcji wobec działania prowadzonego przez organ władzy.
- `upvote/downvote <timestamp> <member> <password> <action>` - jest to akcja związana z głosowaniem przez użytkownika.
- `trolls <timestamp>` – zwraca listę użytkowników, którzy spełniają następujący warunek: spośród wszystkich akcji utworzonych przez użytkownika sumaryczna liczba głosów popierających (*upvotes*) jest mniejsza od liczby głosów przeciwnych (*downvotes*). W proponowanej implementacji zwracane są wszystkie wartości pola *Trolls.down\_upvotes\_count* mniejsze od zera wraz z odpowiadającym *Trolls.member\_id*

## 4. Opis uprawnień użytkowników *init* oraz *app*

1. Użytkownik *init* jest twórcą bazy danych projektu, która wraz z nim istnieje przed uruchomieniem programu (polecenia tworzące go wraz z bazą danych nie należą do kodu programu). Posiada uprawnienia CREATE (do tworzenia tabel), INSERT oraz uprawnienia do tworzenia użytkownika *app* oraz prawo do przekazania mu odpowiednich uprawnień. Użytkownik *init* jest odpowiedzialny wraz z pierwszym uruchomieniem programu z parametrem – *init* za utworzenie tabel zgodnych z diagramem E-R w bazie danych.
2. Użytkownik *app* powinien otrzymać od użytkownika *init* następujące uprawnienia: INSERT, UPDATE, SELECT w ramach wszystkich tabel przynależących do bazy danych projektu. Wraz z drugim uruchomieniem programu jest odpowiedzialny za wykonanie wszystkich funkcji API ze standardowego wejścia w postaci zwracanego wyniku na standardowe wyjście.