

Kamil Breczko

Baza danych

24 października 2017

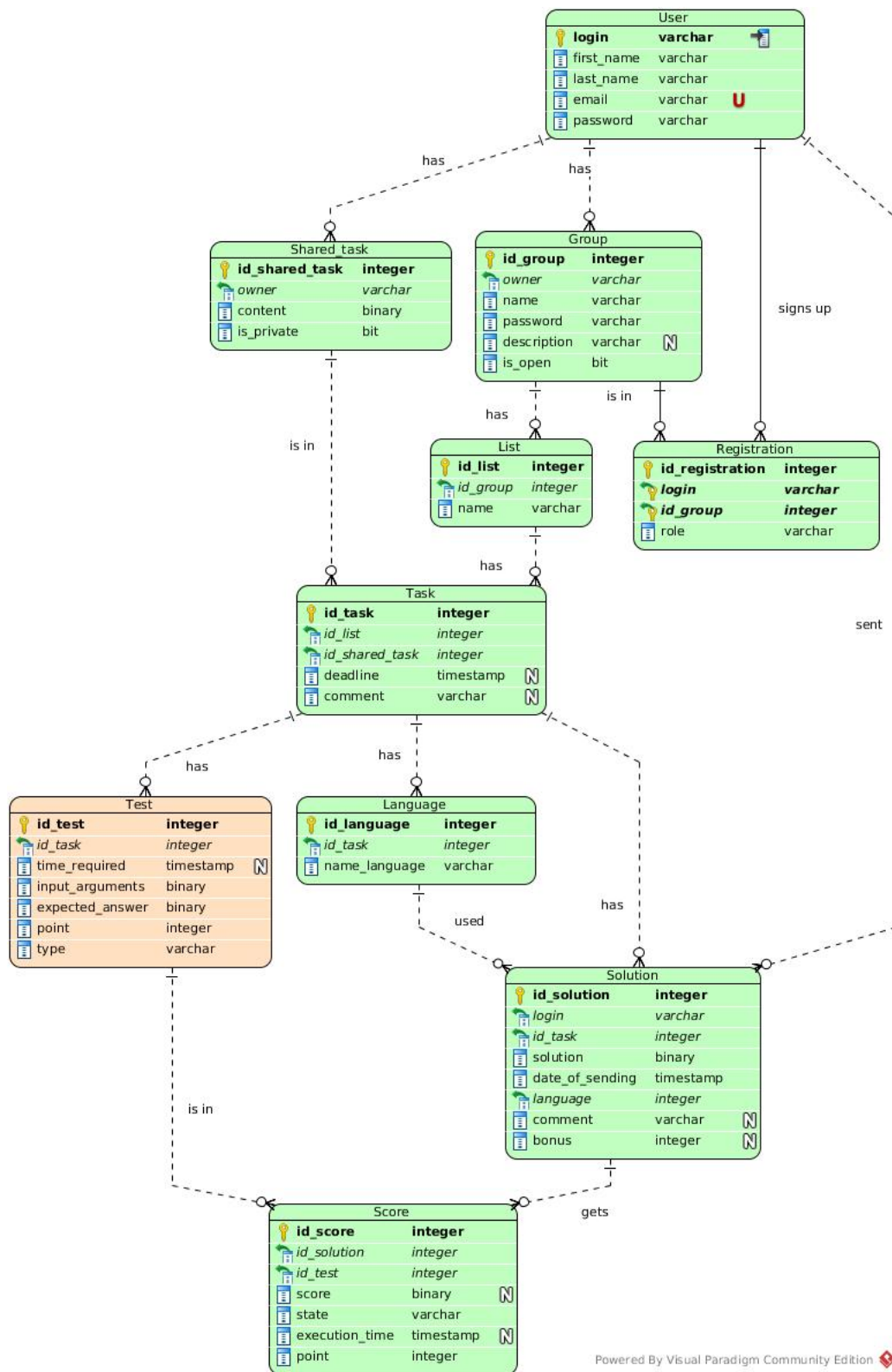
Spis treści

- 1. Encje 3
- 2. Role 4
 - 2.1. Organizator 4
 - 2.2. Użytkownik 4
- 3. Więzy 4
 - 3.1. Encja „Registration” 4
 - 3.2. Encja „Language” 4
 - 3.3. Encja „Test” 4
 - 3.4. Encja „Score” 4

1. Encje

Przy projektowaniu sprawdzarki, potrzebna jest baza danych która posiada informacje o:

- użytkownika;
- grupie, założonej przez organizatora;
- przynależności danego użytkownika do grupy;
- zdefiniowanym zadaniu;
- listach zadań w danej grupie;
- dostępnych językach programowania oraz przydzielonych testach w danym zadaniu;
- wysłanym rozwiązaniu oraz poprawności danego rozwiązania;



Rysunek 1.1. Diagram bazy danych.

Dostęp do encji powinni mieć wszyscy użytkownicy, oprócz encji o nazwie *Test* do której dostęp ma organizator danej grupy.

2. Role

W systemie będą istnieć trzy rodzaje użytkowników: zwykły użytkownik, organizator, administrator.

2.1. Organizator

Uprawnienia organizatora:

- tworzenie grup ćwiczeniowych oraz zadań w danej grupie;
- zarządzanie stworzoną grupą (nadawanie grupie hasła, zmiana nazwy, zamykanie/otwieranie grupy);
- dodawanie list zadań;
- zdefiniowanie zadania;
- możliwość korzystania z stworzonych zadań w innych grupach;
- określenie języków programowania w danym zadaniu;
- wstawienie testów;
- edycja testów;
- publikacja rankingu najlepszych rozwiązań;
- publikacja rankingu najlepszych uczestników w grupie;
- możliwość przydzielania dodatkowych punktów za rozwiązanie;
- komentowanie przesłanego rozwiązania;
- komentowanie zadania w danej grupie;

2.2. Użytkownik

Uprawnienia użytkownika:

- zapis do istniejącej grupy;
- sprawdzenie wyników testów;
- wysyłanie programów, z określonym odstępem w czasie;
- wybór języka programowania;
- wyświetlanie rankingu najlepszych programów w zadaniu (możliwość sortowania wg. czasu i ilości zdobytych punktów za testy);
- przeglądanie starszych wersji rozwiązań;

3. Więzy

3.1. Encja „Registration”

Wartość w kolumnie *role* może przyjmować:

- user;
- organizer;
- administrator;

3.2. Encja „Language”

Wartość w kolumnie *name_language* powinna przyjmować nazwy języków które udostępnia sprawdzarka.

3.3. Encja „Test”

Wartość w kolumnie *type* powinna przyjmować:

- public – użytkownik może wyświetlić dane wejściowe i wyjściowe;
- private – użytkownik może wyświetlić tylko komunikat stanu testu;
- hidden – użytkownik nie widzi testów;

3.4. Encja „Score”

Wartość w kolumnie *state* może przyjmować:

- TLE – Time limit exceeded: program przekroczył limit czasu przewidziany na test;
- RE – Runtime error: program zakończył się z błędem wykonania;
- WA – Wrong answer: wygenerowana przez program odpowiedź jest nieprawidłowa;
- OK – wygenerowana przez program odpowiedź jest poprawna;