

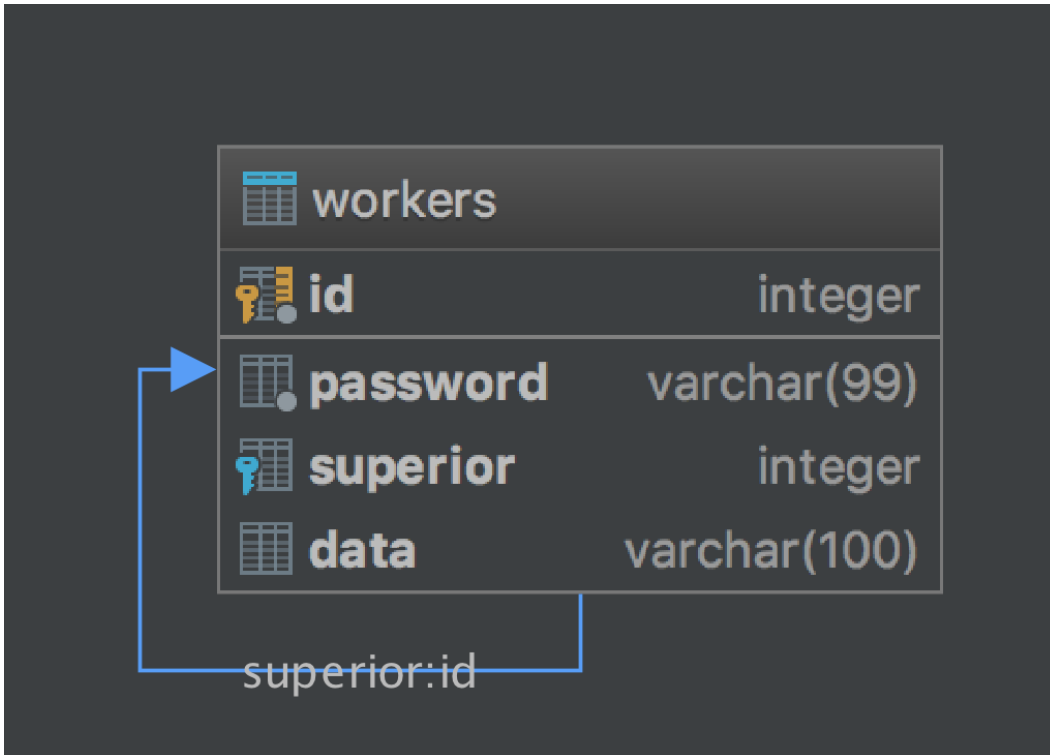
Oskar Sobczyk

Bazy Danych
Dokumentacja projektu

„System Zarządzania Korporacją X”

Wrocław, 11 czerwca 2018

1. Model fizyczny bazy danych



Baza danych składa się z jednej tablicy *workers*. Zawiera ona następujące kolumny:

- id - Unikalny identyfikator każdego pracownika.
- password - Zhaszowane hasło
- superior - Id nadrzędnego przeciwnika
- data - Dane przypisane pracownikowi

2. Autoryzacja

W programie przewidziano dwie metody autoryzacji dostępu. Pierwsza z nich *authorization* pozwala na wykonanie metody dowolnemu pracownikowi firmy niezależnie od jego pozycji w hierarchii. Druga metoda to *authorization hierarchy* pozwala na wykonanie metody tylko użytkownikom którzy znajdują się na ścieżce do danego wierzchołka.

3. Opis klas i funkcji

3. 1. JsonInterpreter

Klasa operuje na pliku JSON zawierającym instrukcje API. Konstruktor klasy wczytuje pierwszą linię pliku zawierającą dane dotyczące połączenia z bazą. Metoda *execute* odpowiada za wczytywanie kolejnych linii pliku i wykonywanie odpowiednich metod klasy *DBInterface*.

3. 2. DBInterface

Klasa odpowiedzialna za obsługę połączenia z bazą danych. Wraz z inicjalizacją klasy następuje nawiązanie połączenia z bazą danych. Opis metod:

- **authenticate** - Metoda odpowiedzialna za autoryzację użytkownika. Autoryzacja zakończy się powodzeniem dla dowolnego użytkownika z tabeli workers.
- **authenticate hierarchy** - Metoda odpowiedzialna za autoryzację użytkownika. Autoryzacja zakończy się powodzeniem jeśli użytkownik <id admin> będzie znajdował się na ścieżce z korzenia do użytkownika <id worker>. Funkcja została zaimplementowana w następujący sposób: zaczynając od <id worker> idzie w górę drzewa (za pomocą danych w kolumnie superior) aż napotka <id admin>. Wtedy kolejne zapytanie sprawdza czy dla <id admin> zostało podane poprawne hasło.
- **initialize** - funkcja odpowiedzialna za inicjalizację bazy danych za pomocą komend zawartych w pliku "init table.sql". Jest ona wykonywana kiedy login do bazy to "init". Utworzy niezbędne tabele, doda użytkownika app i nada mu niezbędne uprawnienia.
- **root i new** - tworzy nowego pracownika. Implementacja opiera się na zapytaniu INSERT które wstawia nowy wiersz do tabeli workers. W przypadku funkcji root parametr Superior ustawiony jest na null.
- **remove** - Funkcja odpowiedzialna za usuwanie pracownika wraz z jego wszystkimi poddrzewami. Dzięki zastosowaniu klucza obcego w komórce Superior oraz reguły "on delete cascade" wystarczy tylko jedno polecenie aby usunąć użytkownika wraz z jego podwładnymi. Przed wykonaniem funkcji sprawdzane jest jeszcze czy usuwany użytkownik nie jest prezesem.
- **child** - Funkcja zwraca listę bezpośrednich podwładnych danego pracownika. Dane te są zdobywane za pomocą zapytania które sprawdza którzy pracownicy mają ustawione pole Superior na wartość id
- **parent** - Funkcja zwraca bezpośredniego przełożonego danego pracownika. Do zdobycia tych informacji wystarczy zapytanie zwracające wartość pola Superior dla danego id.
- **update** - Funkcja zmienia zawartość pola Data dla danego pracownika.
- **read** - Funkcja zwraca wartość pola Data dla danego pracownika.
- **descendants** - Funkcja zwraca wszystkich podwładnych dla danego pracownika. Użyto przeszukiwania metodą DFS. Dla każdego dziecka pracownika są sprawdzane jego dzieci.
- **ancestors** - Funkcja zwraca wszystkich przełożonych pracownika. Została zaimplementowana w sposób w którym sprawdza kto jest bezpośrednim przełożonym pracownika a następnie sprawdza kto jest przełożonym przełożonego aż napotka na korzeń drzewa.
- **ancestor** - Funkcja sprawdza czy jeden pracownik jest przełożonym drugiego. Zaimplementowana została w podobny sposób co ancestors, zaczynając od pierwszego pracownika idzie w górę drzewa aż nie napotka drugiego pracownika którego szukała lub trafi na korzeń drzewa.

4. Uruchamianie programu

4. 1. Wymagania

- Python 3
- Psycopg2
- PostgreSQL

4. 2. Uruchamianie

W celu uruchomienia programu należy wywołać polecenie *python3 main.py <dane.json>* gdzie jako <dane.json> należy podać plik zgodny ze specyfikacją API.