

Bazy danych

Rozproszona Biblioteka

Witold Bołt

20 maja 2006

Spis treści

1	Zawartość dostarczonych plików źródłowych	1
2	Opis ogólny	2
3	Schemat bazy danych	2
3.1	Tablice	2
3.2	Funkcje	5
3.3	Wyzwalacze	6

1 Zawartość dostarczonych plików źródłowych

Projekt składa się kilku plików źródłowych. Poniżej opisano ich zawartość.

- **opis.tex** i **opis.pdf** – opis projektu który czytasz w wersji źródłowej \LaTeX i skompilowanej do formatu PDF (opis zawiera również diagram ERD);
- **erd.pdf** – diagram ERD bazy danych;
- **tabele.sql** – deklaracje tablic (opisane w punkcie 3.1);
- **funkcje.sql** – deklaracje funkcji (opisane w punkcie 3.2);
- **trigger.sql** – deklaracje wyzwalaczy i funkcji w nich wywoływanych (opisane w punkcie 3.3);
- **reguly.sql** – deklaracje reguł (opis można znaleźć przy okazji opisu tablic w 3.1);

- **dane.sql** – przykładowe dane;
- **drop.sql** – plik sprzątający;
- **pytania.sql** – przykładowe zapytania w raz z opisem w komentarzu.

Uwaga. Aby poprawnie załadować bazę danych należy wczytywać pliki SQL w następującej kolejności: **tabele.sql**, **funkcje.sql**, **reguly.sql**, **trigger.sql**, **dane.sql**.

2 Opis ogólny

Prezentowana baza danych została zaprojektowana z myślą o obsłudze systemu zarządzającego tzw. rozproszoną biblioteką. Jej celem jest umożliwienie zarządzania wypożyczeniami książek pomiędzy uczestnikami systemu, którzy stanowią pewną (nie wielką) społeczność, taką jak: studenci jednego roku studiów, pracownicy firmy, uczestnicy kursu czy znajomi z osiedla. Podstawą całego pomysłu jest model w którym każdy użytkownik posiada pewne książki którymi chce się dzielić z innymi. Użytkownicy są podzieleni na jedną lub wiele grup. Użytkownicy mogą prosić o pożyczenie książek od innych użytkowników należących do tej samej grupy (przy czym każdy użytkownik może należeć do kilku grup). System utrzymuje również historię wszystkich wypożyczeń książek w nim zgromadzonych. Pozwala również na wymianę opinii (komentarzy) na temat poszczególnych książek (i konkretnych ich egzemplarzy) między użytkownikami, oraz przypisywanie książek do zdefiniowanych wcześniej kategorii (użytkownicy mają prawo dodawać nowe kategorie).

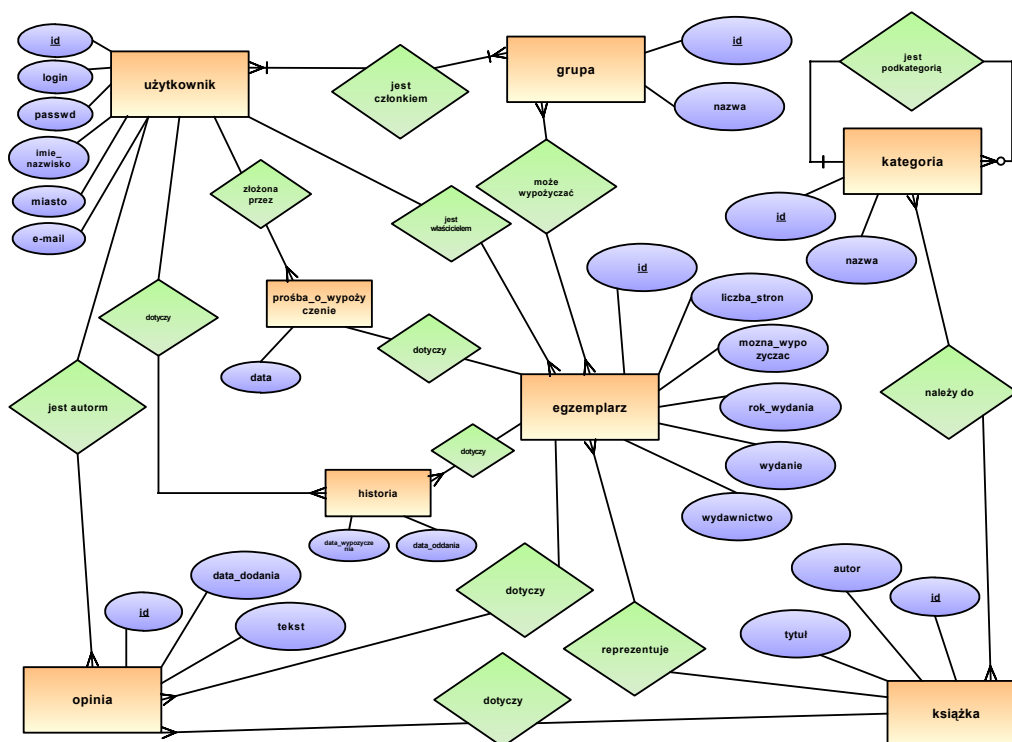
Niniejszy projekt zawiera jedynie opis samej struktury bazy danych która zasila system. Całość powinna być jeszcze oprogramowana za pomocą jakiegoś języka umożliwiającego stworzenie interfejsu dostępu do danych (takiego jak Python, PHP, Java lub C#).

3 Schemat bazy danych

Poniższy diagram przedstawia budowę bazy.

3.1 Tablice

Pokrótkie omówimy teraz tabele występujące w bazie. Ich deklaracje można znaleźć w pliku **tabele.sql**. Reguły które zapewniają spójność danych w tablicach znajdują się w pliku **reguly.sql** (przed ich załadowaniem należy załadować plik z funkcjami, opisany w następnym podrozdziale).



Rysunek 1: Diagram bazy danych.

- **Użytkownik** – zawiera pola, które mogą być wykorzystane przez wyższą warstwę oprogramowania do autoryzacji i identyfikacji użytkowników w systemie.
- **Grupa** – zbiór użytkowników określający uprawnienia odnośnie wypożyczania książek (użytkownicy danej grupy mogą wypożyczać książki od siebie na wzajem – innymi słowy jeśli użytkownik A chce pożyczyć książkę od użytkownika B, to musi istnieć przynajmniej jedna grupa, tak że, zarówno A jak i B należą do niej).
- **Prośba o wypożyczenie** – kataloguje prośby skierowane przez użytkowników odnośnie wypożyczenia książek. Każdy użytkownik znajdujący się w odpowiedniej grupie może złożyć taką prośbę bez względu na to czy dany egzemplarz jest dostępny w danym momencie czy nie (jedynym wyjątkiem jest sytuacja w której znacznik *można wypożyczać* danego egzemplarza ustawiony jest na *fałsz* – zachowanie to kontrolowane jest poprzez odpowiednią regułę). Właściciel egzemplarza jest informowany (przez system wyższej warstwy) o pojawianiu się takiej prośby i może na nią zareagować zgadzając się na wypożyczenie lub nie.

- **Historia** – zawiera historię wszystkich wypożyczeń, a także obecnie trwające wypożyczenia. Jeśli pole *data zwrotu* jest puste to przyjmujemy, że dane wypożyczenie trwa nadal – innymi słowy jeśli chcemy wpisać do systemu informacje o wypożyczeniu danej pozycji – wpisujemy rekord do tabeli historia nie podając daty zwrotu (co więcej jest to jedyny możliwy wariant operacji wstawiania). W bazie danych stworzona jest specjalna perspektywa (widok) o nazwie **wypozyczenia**, która zawiera te rekordy z historii, które mają pustą datę zwrotu. Przyjmuje się również założenie, że jedyną możliwą zmianą w tej tablicy jest dodanie daty zwrotu (zachowania te wymuszone są przez odpowiednie reguły).
- **Opinia** – jest używana do przechowywania opinii o książkach i poszczególnych egzemplarzach. W przypadku opinii o egzemplarzu, może być ona wystawiona tylko przez właściciela lub użytkowników którzy wypożyczyli kiedyś ten egzemplarz (lub mają go obecnie). W przypadku opinii o książkach mogą one być wystawione przez każdego.
- **Egzemplarz** – reprezentuje fizyczną książkę, która znajduje się w posiadaniu kogoś z użytkowników. Pośród kolumn tej tablicy wyjątkową rolę pełni znacznik *można wypożyczać*, który może przyjmować wartość prawda lub fałsz i może być zmieniany przez właściciela danego egzemplarza. Oznacza on dostępność egzemplarza w danym czasie i jest stosowanych w sytuacjach gdy właściciel używa książki w danym momencie, lub wie, że nie może jej nikomu pożyczyć.
- **Książka** – nie oznacza konkretnego fizycznego obiektu (przedmiotu) a jedynie pewien ich zbiór. Każda książka może być przypisana do jednej lub wielu kategorii. Może być reprezentowana przez jeden lub więcej egzemplarzy, przy czym może się zdarzyć tak, że w systemie nie ma żadnego egzemplarza który reprezentowałby daną książkę (np. w momencie dodawania nowych danych do systemu, lub w przypadku zniszczenia, zgubienia, uszkodzenia jedyne istniejącego wcześniej egzemplarza).
- **Kategoria**. Książki przypisane są do różnych kategorii. Kategorie tworzą strukturę drzewiastą – przy czym nie jest to drzewo w ścisłym sensie. Dopuszczamy tu możliwość niespójności (nie ma jednego wyróżnionego korzenia). Nie powinno jednak być żadnych cykli (aby zapewnić taką kontrolę napisano specjalną funkcję w języku PL/SQL, która wywoływana jest poprzez wyzwalacz w momencie wstawiania lub aktualizacji danych o kategoriach).

3.2 Funkcje

W ramach projektu zdefiniowano kilka funkcji, które są używane przez reguły zapewniające spójność i poprawność danych, oraz powinny być używane przez warstwę interfejsu użytkownika. Kod źródłowy funkcji znajduje się w pliku **funkcje.sql** – wszystkie funkcje zostały napisane w języku SQL (poza ostatnią z wymienionych, która napisana jest w języku PL/SQL). Poniżej zamieszczono krótki ich opis:

- **getUserGroups(uid integer)** – funkcja pobiera identyfikator użytkownika i zwraca tabelę zawierającą identyfikatory grup do których należy użytkownik o podanym identyfikatorze.
- **getGroupUsers(gid integer)** – funkcja pobiera identyfikator grupy i zwraca tabelę zawierającą identyfikatory użytkowników którzy należą do grupy o podanym identyfikatorze.
- **getEgzemplarzHistory(eid integer)** – funkcja pobiera identyfikator egzemplarza, a zwraca fragment historii tego egzemplarza w postaci tablicy zawierającej kolumny: *uid* – identyfikator użytkownika, *data_wypozyczenia* oraz *data_zwrotu*.
- **getUserHistory(uid integer)** – funkcja pobiera identyfikator użytkownika, a zwraca fragment historii dotyczący tego użytkownika w postaci tablicy zawierającej kolumny: *eid* – identyfikator egzemplarza, *data_wypozyczenia* oraz *data_zwrotu*.
- **getProsbaForUser(uid integer)** – funkcja pobiera identyfikator użytkownika, zwraca natomiast tablicę zawierającą wszystkie prośby skierowane do tego użytkownika w postaci rekordów: *uid* – identyfikator użytkownika zwracającego się z prośbą, *eid* – identyfikator egzemplarza którego dotyczy prośba, *data* – dokładna data i czas prośby (typ danych *TIMESTAMP*).
- **getPozyczoneByUser(uid integer)** – funkcja pobiera identyfikator użytkownika i zwraca tablicę identyfikatorów egzemplarzy wypożyczonych przez danego użytkownika.
- **getPozyczoneByGroup(gid integer)** – funkcja pobiera identyfikator grupy i zwraca tablicę identyfikatorów egzemplarzy wypożyczonych przez wszystkich użytkowników danej grupy.
- **getEgzemplarzAllowedGroups(eid integer)** – funkcja pobiera identyfikator egzemplarza i zwraca tablicę identyfikatorów grup które mogą wypożyczać dany egzemplarz.

- **getEgzemplarzAllowedUsers(eid integer)** – funkcja pobiera identyfikator egzemplarza i zwraca tablicę identyfikatorów użytkowników którzy mogą wypożyczać dany egzemplarz.
- **mozna_wypożyczyc(eid integer)** – funkcja pobiera identyfikator egzemplarza i zwraca wartość jego znacznika *mozna_wypożyczac* typu *BOOLEAN*.
- **getEgzemplarzByBookUser(kid integer, uid integer)** – funkcja pobiera identyfikator książki i użytkownika, zwraca natomiast tablicę identyfikatorów egzemplarzy danej książki które mogą być wypożyczone przez danego użytkownika.
- **wypożyczenieLen(hid integer)** – funkcja pobiera identyfikator wpisu do historii a zwraca długość danego wypożyczenia w dniach, które jest równa różnicy daty zwrotu i wypożyczenia (o ile książkę już oddano) lub daty obecnej i daty wypożyczenia (jeśli wypożyczenie jeszcze trwa).

3.3 Wyzwalacze

W bazie danych zdefiniowane są 3 wyzwalacze. Każdy z nich sprowadza się do wywołania odpowiedniej funkcji napisanej w języku PL/SQL. Treści tych funkcji jaki i same deklaracje wyzwalaczy można znaleźć w pliku *trigger.sql*. Oto opis funkcji z tego pliku:

- **watchForLoops()** – funkcja używana w wyzwalaczu – sprawdza czy w drzewie kategorii nie występują cykle.
- **fixTree()** – funkcja używana w wyzwalaczu – poprawia drzewo po operacji usuwania.
- **historiaInsertWatch()** – funkcja używana w wyzwalaczu wykonywany przed operacją wstawiania do tabeli **historia**.