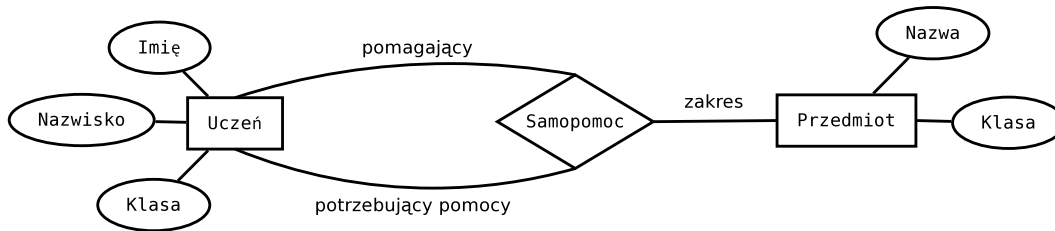


Bazy danych 2013/14. Druga klasówka

Zadanie 1 (15 pkt). Zaproponuj schemat związków encji dla bazy danych przechowującej informacje przydatne w działaniu firmy remontowej. Firma zatrudnia fachowców, którzy potrafią wykonywać różne prace, np. układanie glazury, przerabianie kanalizacji, itp. Fachowcy mogą mieć różne stawki za prace. Bywa, że jeden fachowiec zna się na wielu rzeczach. Klienci składają zamówienia na wykonanie prac, zamówienie może obejmować wiele prac, np. pomalowanie 50 metrów kwadratowych i podłączenie 2 umywalek. Model związków encji powinien umożliwić wycenę zamówienia oraz przypisanie do zamówienia fachowców, którzy zrealizują poszczególne prace. Należy zignorować aspekt czasowy: fachowiec może być przypisany do wielu prac jednocześnie.

Zadanie 2 (10 pkt). Poniższy diagram związków encji przedstawia model danych dla systemu uczniowskich korepetycji w systemie samopomocy.



Dodatkowo wiadmo, że do każdego przedmiotu jest przypisany tylko jeden uczeń pomagający oraz że jeden uczeń może pomagać drugiemu w co najwyżej jednym przedmiocie. Uwaga: romb oznacza związek wielowartościowy; atrybuty są opisane w elipsach, a nie wewnątrz encji.

Zaproponuj realizację tego modelu za pomocą tabel, troszcząc się o to, by

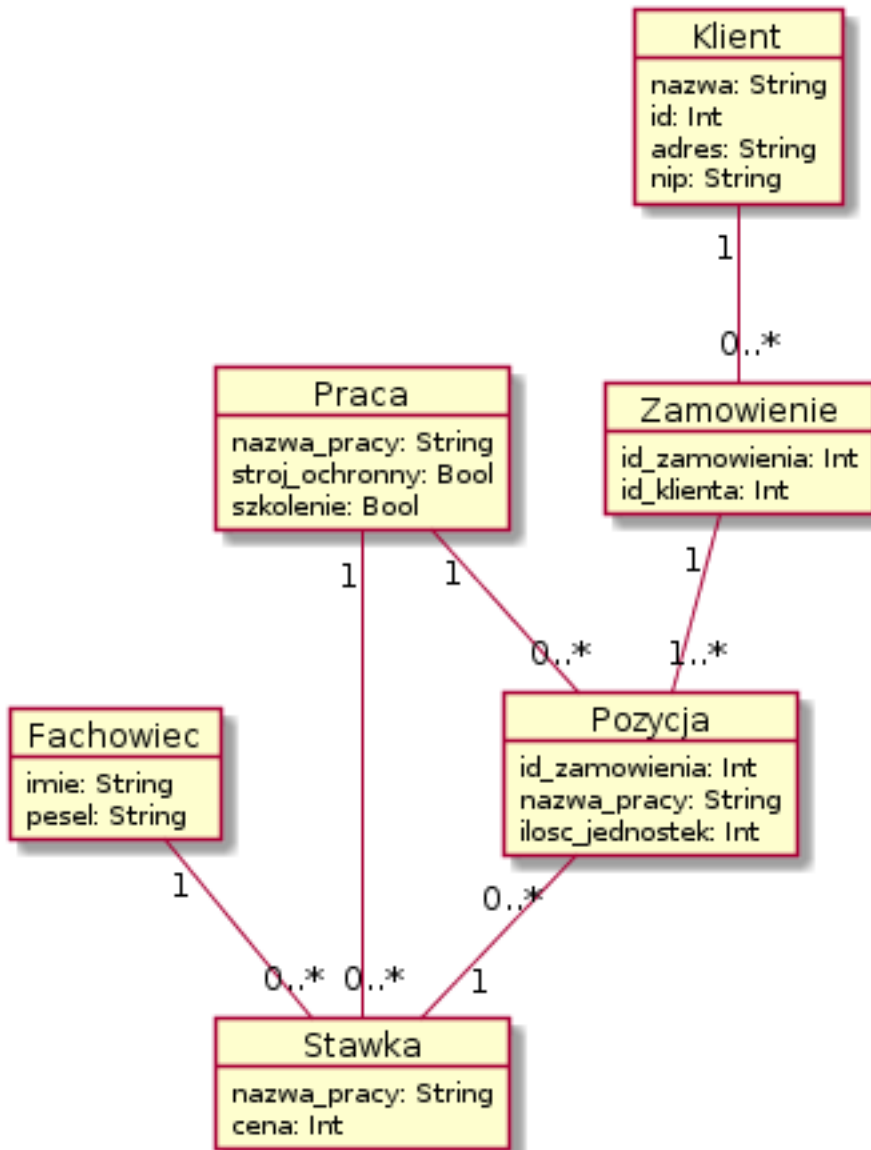
- zależności funkcyjne wynikające z opisu powyżej nie zostały zgubione (wypisz je),
- tabele były w 3NF (uzasadnij, że są),
- tabele były w BCNF, jeśli to możliwe bez utraty zależności funkcyjnych (uzasadnij, że są w BCNF, lub dlaczego nie mogą być w BCNF bez utraty zależności).

Zadanie 3 Przyjmijmy, że dokument `covers.xml` jest zgodny z następującym DTD:

| | |
|---|--|
| <pre><!DOCTYPE Artists[<!ELEMENT Artists (Artist*)> <!ELEMENT Artist (Name, Song*)> <!ELEMENT Name (#PCDATA)> <!ELEMENT Song (#PCDATA)>]></pre> | <pre>Artists → Artist* Artist → Name Song* Name → ε Song → ε</pre> |
|---|--|

- (a) (10 pkt) Napisz zapytanie XQuery lub XPath wypisujące piosenki wykonywane przez co najmniej dwóch wykonawców.
- (b) (* 5 pkt) Zaproponuj inny model dla tych danych (tzn. inne DTD, w dowolnym formalizmie), dla którego łatwiej to samo zapytanie wyrazić w XPath.

Rozwiązanie zadania 1



Rozwiązanie zadania 2

Aby wypisać zależności musimy

1. albo przyjąć, że nazwa+klasa jednoznacznie określają przedmiot, zaś imię+nazwisko+klasa jednoznacznie określają ucznia,
2. albo dołożyć dodatkowe atrybuty iducznia i idprzedmiotu.

Przyjmujemy to drugie rozwiązanie, bo krócej się pisze.

Zależności z tekstu:

zakres → *pomagajacy*
pomagajacy *potrzebujacy* → *zakres*

Tabele

```
CREATE TABLE Uczen (  
  iducznia INTEGER PRIMARY KEY,  
  imie VARCHAR(30) NOT NULL,  
  nazwisko VARCHAR(30) NOT NULL,  
  klasa VARCHAR(4) NOT NULL  
);
```

```
CREATE TABLE Przedmiot (  
  idprzedmiotu INTEGER PRIMARY KEY,  
  nazwa VARCHAR(30) NOT NULL,  
  klasa VARCHAR(4) NOT NULL,  
  pomagajacy INTEGER REFERENCES Uczen  
);
```

```
CREATE TABLE Samopomoc (  
  pomagajacy INTEGER NOT NULL REFERENCES Uczen,  
  potrzebujacy INTEGER NOT NULL REFERENCES Uczen,  
  zakres INTEGER NOT NULL REFERENCES Przedmiot,  
  PRIMARY KEY (pomagajacy, potrzebujacy)  
);
```

Tabele Uczen i Przedmiot są w BCNF, bo mają tylko sztuczne zależności od kluczy.

Tabela Samopomoc ma dwa klucze (pomagajacy, potrzebujacy) i (zakres, potrzebujacy), czyli każdy atrybut jest w kluczu, a więc żadna zależność dla 3NF nie jest naruszona. Dla BCNF naruszona jest pierwsza zależność, ale dalszy rozkład zgubiłby drugą zależność.

Uwagi:

1. Przedmiot musi zawierać pomagającego, bo gdy nikomu nie pomaga, to gubimy zależność (chyba, że nam na tej wiedzy nie zależy).
2. Akceptowane były rozwiązania wybierające wariant 1 (tzn. zakładające wielokolumnowe klucze) z jedną uniwersalną tabelą — pod warunkiem, że reszta była poprawnie.

Rozwiązanie zadania 3

(a) Przykładowe zapytanie XQuery:

```
let $res := distinct-values(  
  for $s1 in /Artists/Artist/Song, $s2 in /Artists/Artist/Song  
  where $s1=$s2 and $s1/../Name!=$s2/../Name  
  return $s1  
)  
return $res
```

(b) DTD może wyglądać np. tak:

```
<!DOCTYPE Songs [  
  <!ELEMENT Songs (Song*)>  
  <!ELEMENT Song (Title, Artist+)>  
  <!ELEMENT Title (#PCDATA)>  
  <!ELEMENT Artist (#PCDATA)>  

```

Przykładowe zapytanie XPath:

```
/Songs/Song[count(Artist)>1]/Title
```