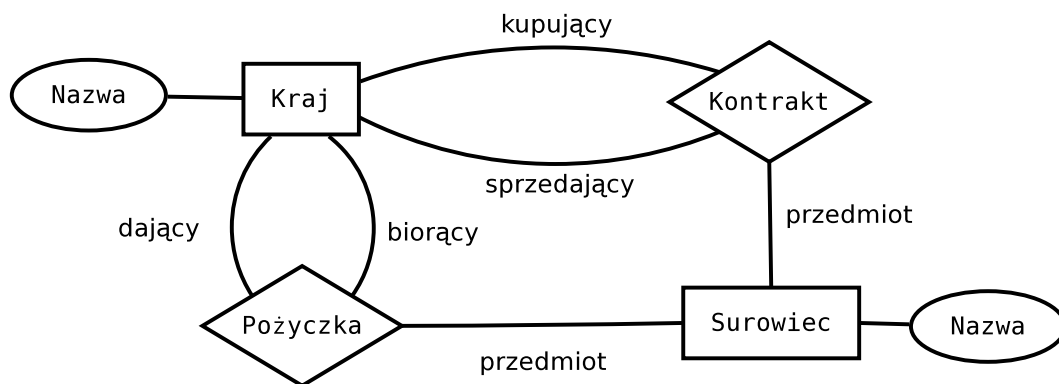


Bazy danych 2014/15. Klasówka z modelowania

Zadanie 1 (15 pkt). Zaproponuj schemat związków encji dla bazy danych przechowującej informacje przydatne w działaniu sklepu handlującego grami (komputerowymi, planszowym, karcianymi, itp.). Gry mają różne rozszerzenia, sprzedawane osobno. Niektóre rozszerzenia wymagają innych rozszerzeń. Kupujący powinien widzieć, które z nich są potrzebne żeby danego rozszerzenia używać. Ponadto sklep ustanawia różnego rodzaju promocje: rabat na komponenty wymagane dla rozszerzeń ostatnio kupionych, rabat na rozszerzenia, których dany klient może już używać (na podstawie jego dotychczasowych zakupów), rabat na drugi egzemplarz. Procentowa kwota rabatu może być zależna od gry/rozszerzenia, której dotyczy. W przyszłości sklep przewiduje również oceny poszczególnych tytułów i wprowadzenie rekomendacji gier, na podstawie podobieństwa innych ocen.

Zadanie 2 (15 pkt). Poniższy diagram związków encji przedstawia model danych dla kontraktów i pożyczek surowców.

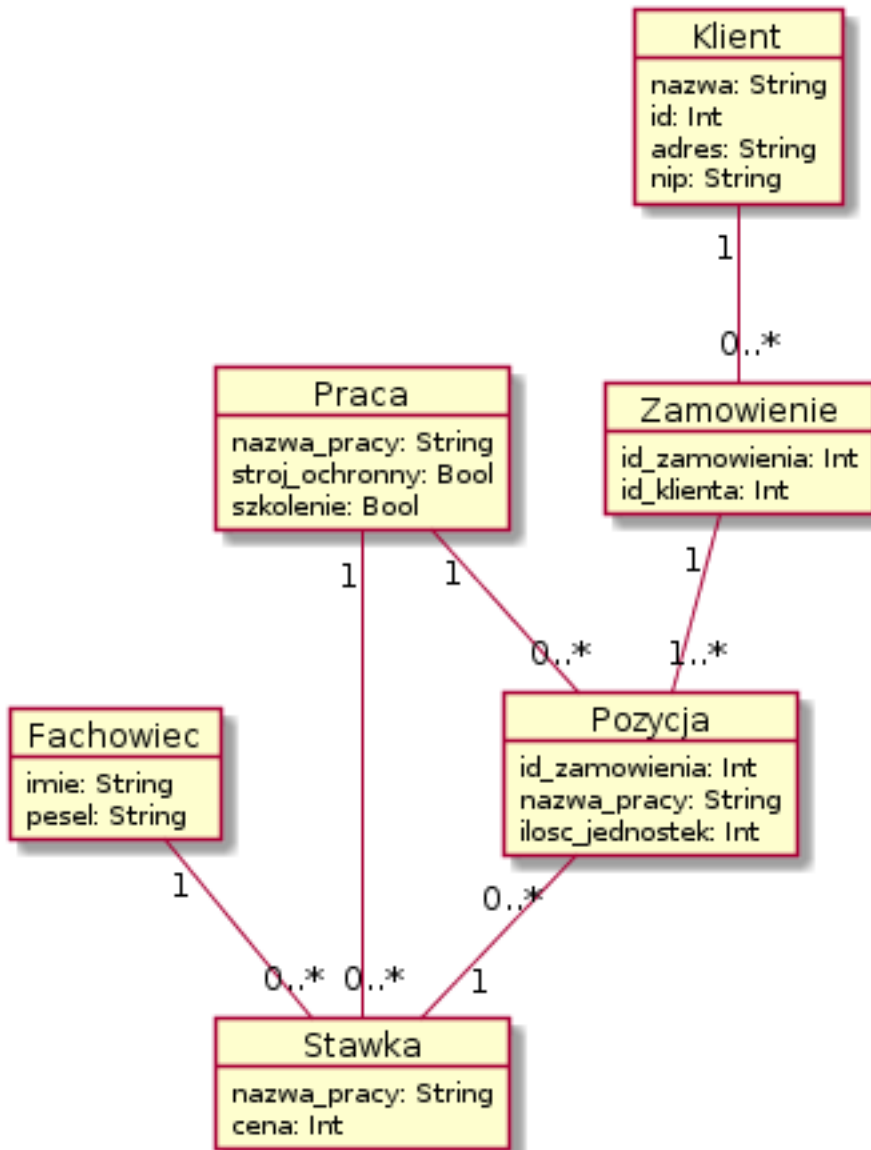


O modelu wiadomo dodatkowo, że w pożyczkach uczestniczą wyłącznie kraje EU i że jeden kraj może pożyczyć innemu krajowi co najwyżej jeden surowiec (jest to regulowane odpowiednim przepisem). Uwaga: romby oznaczają związki o arnościach 3; atrybuty są opisane w elipsach, a nie wewnątrz encji.

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).

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.

Przyjmijmy 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
```