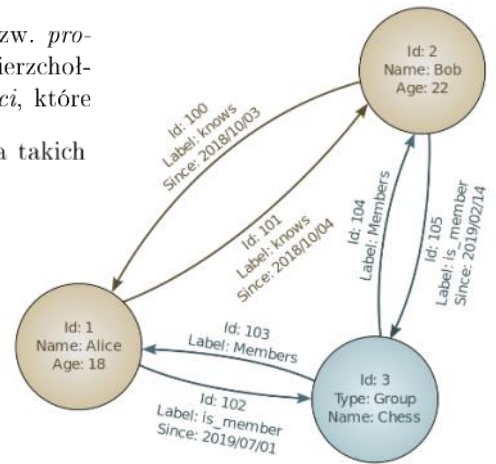


6. (1 pkt) Jednym z najważniejszych obecnie modeli danych dla grafów są tzw. *property graphs* (przykład na rys. 1). Grafy w tym modelu składają się z wierzchołków i krawędzi. Wierzchołki i krawędzie mogą posiadać wiele *własności*, które są parami klucz-wartość. Zaproponuj sensowny sposób przechowywania takich grafów w relacyjnej bazie danych.

7. (1 pkt) Tradycyjnie bardzo rzadko stosowało się postacie normalne powyżej BCNF (np. 4NF). Cała teoria normalizacji powyżej BCNF była uważana za odległą od praktyki.

W dużej mierze wynikało to z faktu, że sposoby wyliczania złączeń w bazach danych zakładały, że operator złączenia jest binarny (łączymy dwie tabele, wynik złączenia z kolejną itd.). W nowoczesnych rozwiązaniach czasami opłaca się posunąć normalizację aż do skrajności - każda tabela posiada co najwyżej jedną kolumnę, która nie jest częścią klucza (tzw. 6. postać normalna, 6NF). Jedną z zalet takiego podejścia jest chęć wykonywania złączeń wielu tabel jednocześnie (*worst-case optimal multi-way joins*).

Zmodyfikuj rozwiązanie poprzedniego zagadnienia tak aby *property graphs* przechowywać w 6NF.



Rysunek 1: Property graph

Pomysł (nieznormalizowany)

Ew. Krawędzie -> edges, properties

Wierzchołki :

δ Id	Name	Age	Type	...
1	Alice	18		
2	Bob	22	Group	
3	Chess			

Krawędzie

Id_from	Id_to	Id	Label	Since
1	2	101		

Przeróbony na 6NF:

Wierzchołki

V_name (Id, name)

V_age (Id, age)

V_type (Id, type)

Krawędzie

edges (Id, Id_from, Id_to)

tutaj pytanie, czy możemy uznać
ze from i to jest dwuczłowym
kluczem? (jak nie to można podzielić)

Własności

e_label (Id, label)

e_since (Id, since)