Bazy danych

3. Zależności funkcyjne

P. F. Góra

http://th-www.if.uj.edu.pl/zfs/gora/

2020

Zależności funkcyjne (ang. *functional dependencies*) to jedno z najważniejszych pojęć teoretycznych w relacyjnym modelu baz danych.

Definicja. Niech $A_1, A_2, \ldots, A_n, B_1, B_2, \ldots, B_m, C_1, C_2, \ldots$ będą atrybutami pewnej tabeli R. Oznaczmy $X = \{A_1, A_2, \ldots, A_n\}$, $Y = B_1, B_2, \ldots, B_m\}$, $Z = \{C_1, C_2, \ldots\}$. Mówimy, że zbiór atrybutów Y *zależy funkcyjnie* od zbioru atrybutów X wtedy i tylko wtedy, gdy każda ustalona wartość X jest jednoznacznie powiązana z dokładnie jedną wartością Y.

Zależności funkcyjne zapisujemy w postaci

$$A_1, A_2, \ldots, A_n \to B_1, B_2, \ldots, B_m$$

lub w skrócie

$$X \to Y$$

Alternatywnie możemy powiedzieć, że wynik rzutowania

$$\pi_{X,Y}(R)$$

określa Y jako funkcję X.

Alternatywnie, jeżeli dwie krotki w tabeli R są zgodne w atrybutach $A_1, \ldots, A_n$ **muszą** być zgodne w atrybutach $B_1, B_2, \ldots, B_m$.

$$X \to Y$$

X	Y	Z
x	y	z_1
x	y	z_2
jeżeli są zgodne tutaj...	...to muszą być zgodne tutaj	

Na przykład spodziewamy się, że w "osobowej" bazie danych obowiązuje zależność funkcyjna PESEL $\to$ Nazwisko. Jeżeli dwie krotki mają taki sam numer PESEL, *muszą* odnosić się do osób mających to samo Nazwisko.

Zauważmy, że odwrotna zależnść funkcyjna w ogólności nie musi zachodzić. W powyższym przykładzie dwie osoby — powiedzmy, matka i córka — mogą mieć to samo nazwisko, ale różne numery PESEL.

- Zależności funkcyjne stanowią więz nałożony na dopuszczalne wartości danych (na przykład *nie wolno* stworzyć dwóch krotek o takich samych PESELach, ale różnych Nazwiskach).
- Zależności funkcyjne “odkrywamy” (lub arbitralnie narzucamy) w procesie analizy tego fragmentu rzeczywistości, który projektowana baza danych ma modelować.
- Zależności funkcyjne należą do schematu bazy.
- Zależności funkcyjne są matematycznym modelem więzów jednoznaczności w modelu relacyjnym baz danych.
- O zależnościach funkcyjnych nie można wnioskować jedynie na podstawie instancji (faktycznych wystąpień) tabel.

Zależności trywialne

- Zależność funkcyjną $A_1, A_2, \ldots, A_n \to B$ nazywam *trywialną*, jeśli atrybut B jest równy któremuś atrybutowi $A_{1,2,\ldots,n}$.

- Jeśli przyjmiemy "skrótowy" zapis zależności z wieloczłonową prawą stroną, zależność jest
  - *Trywialna*, jeśli zbiór złożony z atrybutów B jest podzbiorem zbioru złożonego z atrybutów A
  - *Nietrywialna*, jeśli co najmniej jeden B nie jest A
  - *Całkowicie nietrywialna*, jeśli żadnen B nie jest A.

Reguły wnioskowania (aksjomaty Armstronga)

zwrotność: Jeżeli $\{B_1, B_2, \ldots, B_m\} \subseteq \{A_1, A_2, \ldots, A_n\}$, to $A_1, A_2, \ldots, A_n \to B_1, B_2, \ldots, B_m$.

rozszerzenie: Jeżeli $A_1, A_2, \ldots, A_n \to B_1, B_2, \ldots, B_m$, to $A_1, A_2, \ldots, A_n, C_1, C_2, \ldots, C_k \to B_1, B_2, \ldots, B_m, C_1, C_2, \ldots, C_k$ dla dowolnych $C_1, C_2, \ldots, C_k$.

przechodniość: Jeżeli $A_1, A_2, \ldots, A_n \to B_1, B_2, \ldots, B_m$ oraz $B_1, B_2, \ldots, B_m \to C_1, C_2, \ldots, C_k$, to $A_1, A_2, \ldots, A_n \to C_1, C_2, \ldots, C_k$.

Dodatkowe reguły wnioskowania

Można udowodnić, że poniższe reguły wniokowania wynikają wprost z aksjomatów Armstronga:

- Relacja zwrotna: $X \to X$
- Rozszerzanie: jeżeli $X \to Y$, wtedy $X, Z \to Y$
- Sumowanie: jeżeli $X \to Y$ oraz $X \to Z$, wtedy $X \to Y, Z$
- Rozkład: jeżeli $X \to Y$ oraz $Z \subseteq Y$, to $X \to Z$
- Przechodniość: jeżeli $X \to Y$ oraz $Y \to Z$, to $X \to Z$
- Pseudoprzechodniość: jeżeli $X \to Y$ oraz $Y, Z \to W$, to $X, Z \to W$.

Domknięcia

Niech $\{A_1, A_2, \ldots, A_n\}$ będzie pewnym zbiorem atrybutów, S niech będzie zbiorem zależności funkcyjnych. Domknięciem zbioru $\{A_1, A_2, \ldots, A_n\}$ nad S nazywamy taki zbiór atrybutów B, że jeśli jego elementy spełniają wszystkie zależności funkcyjne z S, to spełniają także zależność $A_1, A_2, \ldots, A_n \to B$, a zatem zależność $A_1, A_2, \ldots, A_n \to B$ wynika z S.

Domknięcie oznaczam cl$\{A_1, A_2, \ldots, A_n\}$.

Mówiąc niezbyt ściśle, domknięcie to zbiór wszystkich atrybutów determinowanych, w sensie zależności funkcyjnych, przez atrybuty zbioru wyjściowego.

Algorytm obliczania domknięcia

1. Na początku X oznacza zbiór $\{A_1, A_2, \ldots, A_n\}$.

2. Znajdujemy wszystkie zależności funkcyjne postaci $B_1, B_2, \ldots, B_m \to C$, gdzie B_i należą do X, a C nie należy. Dołączamy C do X.

3. Powtarzamy krok 2 tak długo, jak długo do X można dołączyć jakiś nowy atrybut. Ponieważ X może się tylko rozszerzać, zaś zbiór atrybutów jest skończony, po skończonej liczbie kroków nastąpi moment, w którym do X nie da się niczego dołączyć.

4. W tym momencie $X = \mathrm{cl}\{A_1, A_2, \ldots, A_n\}$.

Powyżej przedstawiony algorytm jest poprawny i intuicyjnie prosty, ale nie jest efektywny — może być algorytmem kwadratowym (w czasie) w najgorszym przypadku.

Znacznie rozsądniej jest tak uporządkować zależności funkcyjne, aby każda była używana (“odpalana”) dokładnie w tym momencie, w którym wszystkie atrybuty jej lewej strony znajdą się w “kandydacie” X.

Dla dużych baz (i tabel) domknięć nie oblicza się ręcznie! Istnieje specjalny software, który to robi.

Przykład

Rozważmy zbiór atrybutów $\{A, B, C, D, E, F\}$. Załóżmy, że w tym zbiorze zachodzą zależności $A, B \to C,\ B, C \to A, D,\ D \to E,\ C, F \to B$. Obliczmy cl$\{A, B\}$.

$X = \{A, B\}$. Wszystkie atrybuty poprzednika zależności $A, B \to C$ są w X, więc do X dołączamy C. $X = \{A, B, C\}$.

Lewa strona zależności $B, C \to A, D$ jest w X, A jest już w X, więc do X dołączamy D. $X = \{A, B, C, D\}$.

Na mocy zależności $D \to E$, dołączamy do X atrybut E. $X = \{A, B, C, D, E\}$.

Zleżności $C, F \to B$ nie możemy wykorzystać, ponieważ $F \notin X$ i nie ma jak dołożyć F do X.

Ostatecznie cl$\{A, B\} = \{A, B, C, D, E\}$.

Bazy zależności funkcyjnych

Każdy zbiór zależności funkcyjnych pewnego zbioru atrybutów, z którego można wyprowadzić wszystkie inne zależności funkcyjne zachodzące pomiędzy elementami tego zbioru, nazywam bazą zbioru zależności funkcyjnych. Jeśli żaden podzbiór bazy nie jest bazą (nie umożliwia wyprowadzenia wszystkich relacji), bazę tę nazywam bazą minimalną.

Przykład

Mam atrybuty A, B, C i zależności $A \to B$, $A \to C$, $B \to A$, $B \to C$, $C \to A$, $C \to B$. Można teraz wyprowadzić zależności nietrywialne $A, B \to C$, $A, C \to B$, $B, C \to A$ (oraz zależności trywialne). Bazą minimalną jest zbiór $\{A \to B, B \to A, B \to C, C \to B\}$.

Inną bazą minimalną jest $A \to B, B \to C, C \to A$.

Pytanie:

Po co jemy tę żabę?!

Klucze

Mówimy, że zbiór atrybutów $\{A_1, A_2, \ldots, A_n\}$ tworzy klucz pewnej tabeli, jeśli wszystkie pozostałe atrybuty z tej tabeli są funkcyjnie zależne od wskazanego zbioru.

Dwie różne krotki nie mogą mieć tych samych kluczy (jeśli mają takie same klucze, muszą mieć równe także pozostałe atrybuty, a zatem *nie* są różne). Jeżeli przyjmujemy, że tabele są zbiorami krotek — w zabiorze każdy element występuje co najwyżej raz — widzimy, że klucz jednoznacznie identyfikuje krotkę.

Klucz o tej własności, że żaden jego podziór właściwy nie jest kluczem, nazywamy kluczem minimalnym.

Terminologia alternatywna: W niej to, co powyżej nazwaliśmy kluczem minimalnym, nazywa się po prostu kluczem, natomiast każdy nadzbiór klucza nazywamy nadkluczem.

Zauważmy, że zbiór cl$\{A_1, A_2, \ldots, A_n\}$ zawiera wszystkie atrybuty pewnej tabeli wtedy i tylko wtedy, gdy $\{A_1, A_2, \ldots, A_n\}$ jest (nad)kluczem tej tabeli.

Sprawdzenie, czy dany zbiór elementów stanowi klucz tabeli, sprowadza się do sprawdzenia,

- czy wszystkie atrybuty tabeli należą do domknięcia klucza kandydującego,
- czy jakiś właściwy podzbiór klucza kandydującego także nie ma tej właściwości.

Obliczanie domknięć nad zadanym zbiorem zależności funkcyjnych jest formalnym narzędziem służącym do identyfikowania kluczy tabel.