

Normalizacja

Normalizacja schematu bazy danych polega na jego przekształceniu do postaci, w której nie będą występowały następujące anomalie:

1. **redundancja** – powtarzające się dane
2. **anomalie modyfikacji** – zmodyfikowanie wartości atrybutu jednej krotki może wymagać zmodyfikowania wartości atrybutów innych krotek
3. **anomalie usunięcia** – usunięcie zbędnej informacji może prowadzić do usunięcia informacji przydatnych

Ad. 1, 2

Zamówienie

Nr zamówienia	Pozycja	Nazwa towaru	Cena
#1	1	Gogle	70 PLN
#1	2	Narty	260 PLN
#2	1	Gogle	70 PLN

Informacja o cenie gogli niepotrzebnie powtarza się w kilku miejscach (przy założeniu, że **nazwa towaru** jest kluczem). Jeśli chcielibyśmy zmodyfikować ten atrybut, w pierwszej krotce, musielibyśmy dodatkowo zmodyfikować go w krotce trzeciej.

Ad.3

Magazyn

Nazwa towaru	Cena	Nr zamówienia
Gogle	70 PLN	#1
Gogle	70 PLN	#2
Narty	260 PLN	#1

Zakładając, że informacja o towarze znajduje się tylko w relacji **Magazyn**, usunięcie (zbędnej) informacji o zamówieniu #1, powodowałoby usunięcie (przydatnej) informacji o cenie nart.

Dekompozycja

Normalizacja schematu bazy danych polega na **dekompozycji** relacji, w których występują anomalie. Dekompozycja relacji, to rozbięcie jednej relacji na dwie lub więcej relacji. Kiedy dekomponujemy relację $R(A_1, A_2, A_3, \dots, A_n)$ na relacje $T(B_1, B_2, B_3, \dots, B_m)$ i $S(C_1, C_2, C_3, \dots, C_r)$, muszą być zachowane następujące warunki:

1. $(A_1, A_2, A_3, \dots, A_n) = (B_1, B_2, B_3, \dots, B_m) + (C_1, C_2, C_3, \dots, C_r)$ (+ oznacza sumę teoriomnogościową)
2. Krotki w relacji **T** to krotki powstałe przez **rzutowanie** krotek relacji **R** na zestaw atrybutów $(B_1, B_2, B_3, \dots, B_m)$
3. Krotki w relacji **S** to krotki powstałe przez **rzutowanie** krotek relacji **R** na zestaw atrybutów $(C_1, C_2, C_3, \dots, C_r)$

Rzutowanie krotki na zestaw atrybutów polega na **zawężeniu zbioru jej wartości**, tylko do tych,

które odpowiadają **atrybutom, na które krotka jest rzutowana**. W wyniku tej operacji mogą powstać krotki, które **nie różnią się wartością żadnego atrybutu**. Ponieważ instancja danej relacji zawsze jest zbiorem, krotki takie są **utożsamiane**.

Przykład

Relację **Magazyn**, z poprzedniego punktu, możemy zdekomponować na dwie relacje:

Towary(Nazwa towaru, Cena) i **Zamówienia**(Nazwa towaru, Nr zamówienia).

Instancja relacji **Towary** będzie zawierać następujące krotki:

Nazwa towaru Cena

Gogle 70 PLN

Narty 260 PLN

Instancja relacji **Zamówienia** będzie zawierać zaś krotki:

Nazwa towaru Nr zamówienia

Gogle #1

Gogle #2

Narty #1

Odzyskiwanie danych po dekompozycji

Aby **odzyskać instancję relacji**, która została **zdekomponowana**, należy **każdą krotkę** występującą w jednej relacji powstałej po dekompozycji, połączyć z **wszystkimi krotkami** występującymi w drugiej relacji, których **atrybuty wspólne z daną krotką, są identyczne**.

Przykład

Krotki relacji **Towary** i **Zamówienia** mają jeden wspólny atrybut **Nazwa towaru**. Dlatego też dla pierwszą krotkę relacji **Towary** łączymy z pierwszą i drugą krotką relacji **Zamówienia**, otrzymując krotki (Gogle, 70 PLN, #1) i (Gogle, 70 PLN, #2). Drugą krotkę relacji **Towary** łączymy z trzecią krotką relacji **Zamówienia**, otrzymując krotkę (Narty, 260 PLN, #1). W ten sposób otrzymujemy wszystkie krotki występujące w instancji relacji **Magazyn** przed dekompozycją.

Zależności funkcyjne

Mówimy, że F postaci $A_1 A_2 \dots A_n \rightarrow B$ jest **zależnością funkcyjną relacji R** , jeżeli wszystkie krotki **zgodne co do wartości** na atrybutach $A_1 A_2 \dots A_n$ posiadają **tę samą wartość atrybutu B** .

Innymi słowy: istnieje funkcja przeprowadzająca zbiór wektorów wartości atrybutów $A_1 A_2 \dots A_n$ w wartości atrybutu B .

Jeżeli zbiór atrybutów $A_1 A_2 \dots A_n$ określa funkcyjnie więcej niż jeden atrybut B :

- $A_1 A_2 \dots A_n \rightarrow B_1$
- $A_1 A_2 \dots A_n \rightarrow B_2$
- ...
- $A_1 A_2 \dots A_n \rightarrow B_m$

to możemy zastosować zapis skrótowy:

$$A_1 A_2 \dots A_n \rightarrow B_1 B_2 \dots B_m$$

Zależności funkcyjne są własnością schematu bazy danych, a nie konkretnej jej instancji, dlatego też ich wykrycie nie polega na badaniu zawartości bazy danych, lecz badaniu zależności występujących w schemacie.

Występują **trzy typy** zależności funkcyjnych postaci $A_1 A_2 \dots A_n \rightarrow B_1 B_2 \dots B_m$:

- **trywialne** – jeśli zbiór atrybutów $B_1 B_2 \dots B_m$ **jest podzbiorem** zbioru atrybutów $A_1 A_2 \dots A_n$
- **nietrywialne** – jeśli **co najmniej jeden** atrybut B_i **nie należy** do zbioru atrybutów $A_1 A_2 \dots A_n$
- **całkowicie nietrywialne** – jeśli **żaden** z atrybutów B_i **nie należy** do zbioru atrybutów $A_1 A_2 \dots A_n$

Domknięcie zbioru atrybutów

Ważą operacją związaną z zależnościami funkcyjnymi jest obliczanie **domknięcia zbioru atrybutów relacji nad zbiorem zależności funkcyjnych**. Domknięcie zbioru A oznaczamy A^+ .

Niech dany będzie **zbiór atrybutów** $A = (A_1, A_2, \dots A_n)$ oraz zbiór **zależności funkcyjnych** $F = (F_1, F_2 \dots F_n)$. **Domknięciem** zbioru atrybutów A nad zbiorem zależności F nazywamy taki **zbiór atrybutów** B , w którym **dla każdego atrybutu** B_i , należącego do pewnej relacji R spełniającej zależności F , **spełniona jest zależność** $A_1 A_2 \dots A_n \rightarrow B_i$.

Innymi słowy: zbiór B zawiera wszystkie atrybuty zależne funkcyjnie od zbioru atrybutów A .

Algorytm obliczania domknięcia:

1. Niech X oznacza **zbiór domknięcia**. Na początku $X = A$.
2. Dla **każdej zależności** F_i ze zbioru F sprawdzamy, czy **wszystkie atrybuty** stojące po **lewej stronie** tej zależności **należą do zbioru** X . Jeśli tak, to do zbioru tego **dodajemy wszystkie atrybuty stojące po prawej stronie zależności** F_i (o ile wcześniej nie występowały w tym zbiorze).
3. Poprzedni krok powtarzamy **tak długo, aż zbiór** X **przestanie się powiększać**.
4. Zbiór X **stanowi domknięcie** zbioru A nad zbiorem zależności F .

Przykład

Niech dana będzie relacja $R(A, B, C, D, E, F)$ oraz zbiór zależności funkcyjnych F spełnianych przez tę relację: $A C \rightarrow E, B \rightarrow C, A E \rightarrow D, F \rightarrow A$ (pomijamy zależności trywialne). Należy obliczyć domknięcie zbioru atrybutów (A, B) nad zbiorem zależności F :

1. $X_0 = (A, B)$
2. W pierwszej iteracji do zbioru X możemy dołączyć atrybut C , ponieważ B należy do X_0

- oraz $B \rightarrow C$ należy do F. $X_1 = (A, B, C)$
3. W drugiej iteracji do zbioru D możemy dołączyć atrybut E, ponieważ A i C należą do X_1 oraz $A C \rightarrow E$ należy do F. $X_2 = (A, B, C, E)$
 4. W trzeciej iteracji do zbioru X możemy dołączyć atrybut D, ponieważ A i E należą do X_2 oraz $A E \rightarrow D$ należy do F. $X_3 = (A, B, C, E, D)$.
 5. Widzimy, że do zbioru X nie można dodać już żadnego atrybutu, zatem algorytm kończy działanie.

$$(A, B)^+ = (A, B, C, E, D)$$

Odtwarzanie zależności funkcyjnych po dekompozycji

Niech dane będą: **relacja R**, **relacja S**, która powstała w wyniku dekompozycji relacji R i **zbiór zależności funkcyjnych F**, spełnianych w relacji R. Żeby określi **zbiór zależności funkcyjnych spełnianych w S** należy:

1. Rozważyć **wszystkie podzbiory X zbioru atrybutów relacji S**. Dla każdego z nich określić domknięcie X^+ nad zbiorem zależności F.
2. Jeśli jakiś atrybut **B należący do S** spełnia następujące warunki:
 1. **należy do X^+**
 2. **nie należy do X**
3. to zależność funkcyjna $X \rightarrow B$ **jest spełniona w S**

Klucze relacji

Mówimy, że atrybut lub zbiór atrybutów $A_1 A_2 \dots A_n$ tworzy **klucz relacji R**, jeżeli spełnione są warunki:

- wszystkie atrybuty relacji R **zależą funkcyjnie** od tego atrybutu (zbioru atrybutów)
- (w przypadku gdy klucz jest zbiorem atrybutów) nie istnieje **żaden podzbiór właściwy** zbioru atrybutów $A_1 A_2 \dots A_n$, od którego wszystkie atrybuty relacji R zależą funkcyjnie

Jeśli chcemy sprawdzić czy dany zbiór atrybutów jest kluczem, musimy najpierw zbadać, czy wszystkie atrybuty danej relacji zależą funkcyjnie od tego zbioru atrybutów (obliczając domknięcie danego zbioru atrybutów, nad zależnościami funkcyjnymi występującymi w tej relacji), a następnie sprawdzić, czy usunięcie któregośkolwiek atrybutu ze zbioru, prowadzi do niespełnienia tej własności.

Nadklucz relacji

Mówimy, że zbiór atrybutów $A_1 A_2 \dots A_n$ relacji R jest **nadkluczem** tej relacji, jeżeli **zawiera klucz** (lub co najmniej jeden z kluczy) tej relacji.

Postać normalna Boyce'a-Codda (BCNF)

Mówimy, że relacja R jest w **postaci normalnej Boyce'a-Codda**, jeśli we wszystkich występujących w niej **nietrzywalnych zależnościach funkcyjnych** po lewej stronie występuje **nadklucz tej relacji**.

Aby przekształcić relację nie spełniając warunku BCNF należy dokonać jej **dekompozycji** na szereg relacji $R_1, R_2 \dots R_n$, z których każda spełnia warunek BCNF. Poszczególne etapy dekomponowania polegają na:

1. Sprawdzeniu, czy w zbiorze otrzymanych relacji nie występuje **relacja łamiąca warunek BCNF**. Jeśli nie, to algorytm kończy działanie.
2. Dla każdej relacji łamiącej warunek BCNF należy zidentyfikować **zależność funkcyjną, która powoduje niespełnienie tego warunku** (tzn. nietrzywalnej zależność postaci: $A_1 A_2 \dots A_n \rightarrow B_1 B_2 \dots B_m$, w której $A_1 A_2 \dots A_n$ nie jest nadkluczem tej relacji)
3. **Zdekomponować wszystkie relacje łamiące warunek BCNF** do dwóch relacji zawierających:
 1. atrybuty $A_1 A_2 \dots A_n B_1 B_2 \dots B_m$
 2. atrybuty $A_1 A_2 \dots A_n$ oraz atrybuty nie należące do zbioru $B_1 B_2 \dots B_m$
4. **Powrocie** do pierwszego punktu algorytmu.

Przykład

Niech $R(A, B, C, D, E)$ będzie relacją, w której spełnione są zależności funkcyjne $A B C \rightarrow D$, $A \rightarrow E$. Kluczem relacji jest zbiór atrybutów (A, B, C) . Zależność $A \rightarrow E$ łamie warunek BCNF. Relacja R powinna zostać zdekomponowana na relacje $S(A, B, C, D)$ oraz $T(A, E)$.

W relacji S zachodzi zależność funkcyjna $A B C \rightarrow D$, ponieważ $(A, B, C)^+ = (A, B, C, D)$ i D nie należy do (A, B, C) .

W relacji T zachodzi zależność funkcyjna $A \rightarrow E$, ponieważ $(A)^+ = (A, E)$ i E nie należy do (A) .

Trzecia postać normalna (3NF)

Warunek **trzeciej postaci normalnej (3NF)** jest nieznacznym osłabieniem warunku BCNF.

Mówimy, że relacja R jest w **trzeciej postaci normalnej**, jeśli we wszystkich występujących w niej nietrzywalnych zależnościach funkcyjnych **po lewej występuje nadklucz tej relacji lub po prawej stronie występuje atrybut, który jest elementem pewnego klucza**.

Zależności wielowartościowe

Zależności wielowartościowe są **rozszerzeniem** pojęcia **zależności funkcyjnych**, tzn. każda zależność funkcyjna jest również zależnością wielowartościową, ale nie każda zależność wielowartościowa jest zależnością funkcyjną.

Zbadanie występowania zależności wielowartościowych pozwala na wykrycie redundancji, które

mogą pojawić się w bazie danych, która jednak spełnia warunek BCNF.

Niech R będzie relacją składającą się z trzech grup atrybutów:

1. $A_1 A_2 \dots A_n$
2. $B_1 B_2 \dots B_m$
3. $C_1 C_2 \dots C_l$

Mówimy, że $A_1 A_2 \dots A_n \twoheadrightarrow B_1 B_2 \dots B_m$ jest **zależnością wielowartościową** występującą w relacji R , jeśli dla **każdej pary różnych krotek t i u** , które mają takie same wartości atrybutów A_i można **znaleźć krotkę v** , której składowe mają wartości równe:

1. wartościom atrybutów A_i w krotkach t i u
2. wartościom atrybutów B_i krotki t
3. wartościom atrybutów C_i krotki u

Zależność wielowartościową $A_1 A_2 \dots A_n \twoheadrightarrow B_1 B_2 \dots B_m$ występującą w relacji R nazywamy **nietrywialną**, gdy:

1. **żaden** atrybut B_i nie występuje po **lewej stronie** zależności
2. atrybuty A_i, B_i **nie obejmują wszystkich atrybutów** występujących w relacji R

Przykład

Przypuśćmy, że w relacji **Osoby** poza imieniem i nazwiskiem poszczególnych osób, przechowywane są informacje o posiadanych przez nie samochodach i mieszkaniach. Jedna osoba może mieć zarówno wiele mieszkań, jak i wiele samochodów.

Instancja tej relacji mogłaby wyglądać następująco:

Imię	Nazwisko	Miasto	Adres	Marka samochodu
------	----------	--------	-------	-----------------

Jan	Kowalski	Gdańsk	Mroźna 1	Ford KA
-----	----------	--------	----------	---------

Jan	Kowalski	Gdańsk	Mroźna 1	Opel Astra III
-----	----------	--------	----------	----------------

Jan	Kowalski	Łódź	Prosta 4	Ford KA
-----	----------	------	----------	---------

Jan	Kowalski	Łódź	Prosta 4	Opel Astra III
-----	----------	------	----------	----------------

Redundancja widoczna jest na pierwszy rzut oka, tym niemniej relacja ta spełnia warunek BCNF (kluczem jest pełny zestaw atrybutów – należy wziąć pod uwagę, że w jednym mieście jedna osoba może posiadać wiele mieszkań), ponieważ nie występuje w niej żadna nietrywialna zależność funkcyjna. W relacji tej występują jednak dwie nietrywialne zależności wielowartościowe:

1. Imię Nazwisko \twoheadrightarrow Miasto Adres
2. Imię Nazwisko \twoheadrightarrow Marka samochodu

Czwarta postać normalna (4NF)

Mówimy, że relacja R jest w **czwartej postaci normalnej (4NF)**, gdy dla **każdej, nietrywialnej zależności wielowartościowej** postaci $A_1 A_2 \dots A_n \twoheadrightarrow B_1 B_2 \dots B_m$, **lewa strona jest**

nadkluczem w R.

Normalizacja relacji do postaci 4NF polega na wykryciu zależności wielowartościowej, która łamie warunek 4NF i zdekomponowaniu jej podobnie jak ma to miejsce w przypadku postaci BCNF.

Przykład

W przykładzie z poprzedniego punktu zbiór atrybutów (Imię, Nazwisko) nie stanowi nadklucza relacji Osoby, a występuje w dwóch zależnościach wielowartościowych. Konieczne jest zatem zdekomponowanie tej relacji do dwóch relacji:

1. Adresy(Imię, Nazwisko, Miasto, Adres)
2. Samochody(Imię, Nazwisko, Marka samochodu)

