Projektowanie Baz Danych
Zależności funkcyjne, postacie normalne i normalizacja

Antoni Ligęza

ligeza@agh.edu.pl
http://galaxy.uci.agh.edu.pl/~ligeza

Przykład 1 – błędny projekt schematu tabeli

Naruszenie 2 NF – Tabela DostawcyTowary

dostawca	adres	towar	cena
Budex	Kraków	cegła	1.20
Budex	Kraków	pustak	3.20
Murex	Tarnów	cegła	1.10
Murex	Tarnów	pustak	2.90
Budex	Kraków	papa	4.30

Problemy

- Redundancja – niepotrzebnie powtarzany adres dostawcy,
- Anomalie przy modyfikacji – adres zmodyfikowany w jednym wierszu może pozostać niezmieniony w innych,
- Anomalie przy wstawianiu – nie można wstawić dostawcy bez towaru,
- Anomalie przy usuwaniu – usuwając informację o towarach można usunąć dostawcę.

Zależności funkcyjne:

$$dostawca \longrightarrow adres$$

$$dostawca, towar \longrightarrow cena$$

Klucz: $\{dostawca, towar\}$

Przykład 2 – błędny projekt schematu tabeli

Naruszenie 3NF – Tabela PracownicyFirmy

```
id_prac  |  nazwisko  | firma | adres
---------+------------+-------+--------
 MF101   | Kowalski   | Budex | Kraków
 MF102   | Kowalowski | Budex | Kraków
 MF103   | Maliniak   | Murex | Tarnów
 MF104   | Malinowski | Murex | Tarnów
 MF105   | Mamoń      | Budex | Kraków
```

Problemy

- Redundancja – niepotrzebnie powtarzany adres firmy,
- Anomalie przy modyfikacji – adres zmodyfikowany w jednym wierszu może pozostać niezmieniony w innych,
- Anomalie przy wstawianiu – nie można wstawić firmy bez pracownika,
- Anomalie przy usuwaniu – usuwając wszystkich pracowników usuwamy firmę.

Zależności funkcyjne:

$$id\_prac \longrightarrow nazwisko$$

$$id\_prac \longrightarrow firma$$

$$firma \longrightarrow adres$$

Klucz: $\{id\_prac\}$

Zależności funkcyjne

Definicja 1 *Niech*

$$R(A_1, A_2, \ldots, A_n)$$

będzie schematem pewnej relacji , oraz niech

$$X, Y \subseteq \{A_1, A_2, \ldots, A_n\}.$$

Istnieje zależność funkcyjna (X wyznacza funkcyjnie Y):

$$X \longrightarrow Y$$

wtedy i tylko wtedy, gdy dla dowolnej relacji T o schemacie $R(A_1, A_2, \ldots, A_n)$, dla dowolnych krotek $t_1, t_2 \in T$ jeżeli

$$\pi_X(t_1) = \pi_X(t_2)$$

to także

$$\pi_Y(t_1) = \pi_Y(t_2)$$

Zależność funkcyjna pomiędzy zbiorami atrybutów X oraz Y oznacza, że każdemu zestawowi wartości atrybutów X odpowiada dokładnie jeden zestaw wartości atrybutów Y.

Zależność trywialna:

$$X \longrightarrow Y$$

gdy $Y \subseteq X$.

Zależność prosta (prawa strona jest pojedynczym atrybutem):

$$X \longrightarrow A_i$$

Zależność tranzytywna (przechodnia):

$$X \longrightarrow Y$$

$$Y \longrightarrow Z$$

gdy nie zachodzi $Y \longrightarrow X$.

Przykłady zależności funkcyjnych

Rozkład zajęć uczelni

Rozważmy relację R opisującą rozkład zajęć uczelni o schemacie:

$$R(D, H, S, G, P, C)$$

gdzie:

- D – dzień,
- H – godzina,
- S – sala,
- G – grupa studencka,
- P – prowadzący zajęcia,
- C – przedmiot/kurs.

W powyższym schemacie istnieją zależności funkcyjne, np.:

$$D, H, S \longrightarrow G, P, C$$

$$D, H, G \longrightarrow S, P, C$$

$$D, H, P \longrightarrow S, G, C$$

W konkretnym systemie mogą też istnieć inne zależności, np.

$$D, H, C \longrightarrow S, G, P$$

$$P \longrightarrow C.$$

Własności zależności funkcyjnych

- zależności funkcyjne są własnością **schematu relacji** a nie relacji jako takiej (danych),
- zależności funkcyjne mogą być obiektywne (wynikać z własności systemu) lub subiektywne (z założeń, wymagań, przepisów),
- istnienia zależności funkcyjnych nie dowodzi się; można je zaobserwować lub założyć,
- wybór zależności może być arbitralny; należy do eksperta,
- wybór zależności ma wpływ na projekt bazy danych,
- nie uwzględnienie zależności funkcyjnych może prowadzić do problemów (redundancji, anomalii),
- zależności funkcyjne mogą implikować inne zależności funkcyjne,
- wskazane jest ograniczenie zbioru zależności poprzez eliminację zależności nadmiarowych do ograniczonego zbioru F.

1. Rozważamy tylko zależności gdzie prawa strona jest pojedynczym atrybutem postaci $X \longrightarrow A$,
2. Ze zbioru zależności F nie można usunąć żadnej zależności – $F \setminus \{X \longrightarrow A\}$ nie jest równoważne F,
3. Każda zależność zbioru F jest *pełna*, tzn. dla każdego $Z \subset X$, $F \setminus \{X \longrightarrow A\} \cup \{Z \longrightarrow A\}$ nie jest równoważne F

Domknięcie tranzytywne zbioru atrybutów

Definicja 2 *Domknięciem tranzytywnym zbioru atrybutów U względem zbioru (przy danych) zależności funkcyjnych F nazywamy zbiór U^+:*

$$U^+ = \{A \in U : U \longrightarrow A\}$$

Algorytm generacji domknięcia tranzytywnego

1. Dane: U, F.
2. $U_0 = U$.
3. $U_{i+1} = U_i \cup \{A : X \longrightarrow Y \in F, X \subseteq U_i, A \in Y\}$.
4. Jeżeli $U_{i+1} = U_i$ to stop; $U^+ = U_i$.

Powyższy algorytm umożliwia generację zbioru U^+ – wszystkich atrybutów zależnych funkcyjnie od U.

Zastosowania:

1. Umożliwia znalezienie wszystkich atrybutów zależnych funkcjonalnie od U.
2. Umożliwia generację wszystkich zależności prostych postaci $U \longrightarrow A$.
3. Umożliwia sprawdzenie czy danych zbiór U jest kluczem (nadkluczem).
4. Umożliwia generację wszystkich kluczy (podział na atrybuty kluczowe i niekluczowe).
5. Umożliwia sprawdzenie, czy dana zależność funkcyjna jest pełna (od całej lewej strony).

Logiczne implikacje zależności funkcyjnych i Aksjomaty Armstronga

Definicja 3 *Zależność funkcyjna* $X \longrightarrow Y$ *jest logiczną konsekwencją zbioru zależności* F*, co notujemy*

$$F \models X \longrightarrow Y,$$

wtedy i tylko wtedy, gdy każdy schemat relacji R *w którym zachodzą zależności* F *spełnia również* $X \longrightarrow Y$*.*

Aksjomaty Armstronga

A1 – Zwrotność: Jeżeli $Y \subseteq X \subseteq U$ to $X \longrightarrow Y$ jest logiczną konsekwencją F.

A2 – Rozszerzanie: Jeżeli $X \longrightarrow Y$ oraz $Z \subseteq U$ to zachodzi $X, Z \longrightarrow Y, Z$.

A3 – Przechodniość: Jeżeli $X \longrightarrow Y$ oraz $Y \longrightarrow Z$ to zachodzi $X \longrightarrow Z$.

Aksjomaty Armstronga stanowią **pełny** zbiór reguł i pozwalają wygenerować wszystkie logiczne konsekwencje zależności funkcyjnych F^+, tzn. domknięcie przechodnie zbioru F. Twierdzenia wynikające z powyższych aksjomatów:

- Reguła sumowania: $\{X \longrightarrow Y, X \longrightarrow Z\} \models X \longrightarrow Y, Z$;
- Reguła pseudoprzechodniości: $\{X \longrightarrow Y;\ W, Y \longrightarrow Z\} \models X, W \longrightarrow Z$;
- Reguła rozkładania: Jeżeli $\{X \longrightarrow Y\}$ oraz $Z \subseteq Y$ to $X \longrightarrow Z$;

Twierdzenie 1

$$(F \models X \longrightarrow Y) \equiv (Y \subseteq X^+)$$

Rozkład relacji

Definicja 4 *Niech*

$$U = \{A_1, A_2, \ldots, A_n\}$$
$$A, B, C \subseteq U.$$

Rozkładem relacji R o schemacie $R(A)$ na dwie relacje $R_1(B)$ oraz $R_2(C)$ nazywamy zastąpienie relacji R relacjami R_1 oraz R_2 takimi, że

$$R_1 = \pi_B(R), \quad R_2 = \pi_C(R),$$
$$B \cup C = A.$$

Rozkład powinien zachowywać:

- informację: dla każdej relacji R

$$\pi_B(R) * \pi_C(R) = R,$$

- zależności funkcyjne:

$$(\pi_B(F) \cup \pi_C(F))^+ = F^+$$

gdzie

$$\pi_Z(F) = \{X \longrightarrow Y \in F : X \cup Y \subseteq Z\}$$

Twierdzenie 2 *Rozkład relacji R na R_1 oraz R_2 zachowuje informację wtedy i tylko wtedy gdy*

$$(R_1 \cap R_2) \longrightarrow (R_1 \setminus R_2) \in F^+$$

lub

$$(R_1 \cap R_2) \longrightarrow (R_2 \setminus R_1) \in F^+.$$

Uwagi

Racjonalne warunki rozkładu relacji

- zachowanie wszystkich atrybutów,
- nie rozkładanie z naruszeniem zależności funkcyjnych,
- rozkładanie z wydzieleniem zależności funkcyjnej.

Przykład 1:

$U = \{D, A, T, C\}$, $F = \{D \longrightarrow A; D, T \longrightarrow C\}$
$R_1 = \{D, A\}$, $R_2 = \{D, T, C\}$.

Sprawdzenie: $R_1 \setminus R_2 = \{A\}$, $R_1 \cap R_2 = \{D\}$, $D \longrightarrow A \in F$; rozkład zachowuje informację i zależności funkcyjne.

Przykład 2:

$U\{ID, N, F, A\}$, $F = \{ID \longrightarrow N; ID \longrightarrow F; F \longrightarrow A\}$.
$R_1 = \{ID, N, F\}$, $R_2 = \{F, A\}$.

Sprawdzenie: $R_2 \setminus R_1 = \{A\}$, $R_1 \cap R_2 = \{F\}$, $F \longrightarrow A \in F$; rozkład zachowuje informację i zależności funkcyjne.

Przykład 3:

$U = \{M, U, K\}$, $F = \{M, U \longrightarrow K; K \longrightarrow M\}$

Brak możliwości sensownego rozkładu relacji.

Postacie normalne – 1NF, 2NF i 3NF

Pierwsza postać normalna – 1NF

Definicja 5 *Relacja jest w pierwszej postaci normalnej (1NF) wtedy i tylko wtedy gdy wszystkie dane są atomiczne (regularna tablica prostokątna, na przecięciu każdego wiersza i kolumny – dana atomiczna).*

Relacja dająca sie reprezentować w postaci tabeli bazodanowej w zadzie jest w pierwszej postaci normalnej.

Druga postać normalna – 2NF

Definicja 6 *Relacja jest w drugiej postaci normalnej (2NF) wtedy i tylko wtedy gdy jest w pierwszej postaci normalnej oraz wszystkie* **atrybuty niekluczowe** *są w* **pełni funkcjonalnie zależne** *od wszystkich kluczy.*

Naruszenie drugiej postaci normalnej wymaga istnienia klucza złożonego i atrybutu niekluczowego; atrybut taki musi być zależny funkcyjnie od fragmenty tego klucza.

Trzecia postać normalna – 3NF

Definicja 7 *Relacja jest w trzeciej postaci normalne (3NF) wtedy i tylko wtedy gdy jest w drugiej postaci normalnej (2NF) oraz wszystkie* **atrybuty niekluczowe** *są w* **bezpośrednio zależne** *od wszystkich kluczy.*

Naruszenie trzeciej postaci normalnej wymaga istnienia zależności tranzytywnej od klucza dla atrybutu niekluczowego.

Postacie normalne – BCNF

Postać normalna Boyce'a-Codda– BCNF

Definicja 8 *Relacja jest w postaci normalnej Boyce'a-Codda (BCNF) wtedy i tylko wtedy gdy wszystkie zależności kluczowe są zależnościami od kluczy.*

Naruszenie trzeciej postaci normalnej wymaga istnienia zależności funkcyjnej tranzytywnej lub od fragmentu klucza.

Każda relacja daje sie sprowadzić do 3NF i często do BCNF; relacja z przykładu 3 jest w 3NF ale nie w BCNF.

Algorytm normalizacji

Postępowanie normalizacyjne (lub sprawdzenie) przeprowadzamy dla każdej tablicy RBD.

1. Identyfikacja zależności funkcyjnych – zbiór F.
2. Generacja wszystkich kluczy; podział atrybutów na kluczowe i niekluczowe.
3. Sprawdzenie 2NF: czy są atrybuty niekluczowe zależne od fragmentu klucza? Jeżeli tak – normalizacja przez dekompozycję relacji.
4. Sprawdzenie 3NF: czy są atrybuty niekluczowe zależne tranzytywnie od klucza? Jeżeli tak – normalizacja poprzez dekompozycje relacji.
5. Sprawdzenie BCNF – czy są zależności nie od klucza? Jeżeli tak *próba* dekompozycji.
6. Analiza denormalizacyjna.

Złote rady

Rozpoznawanie 2NF i 3NF

2NF – warunki wystarczające:

- istnieją tylko klucze proste,
- wszystkie atrybuty są kluczowe (brak atrybutów niekluczowych.

3NF – warunki wystarczajace:

- wszystkie atrybuty są kluczowe (brak atrybutów niekluczowych,
- brak zależności tranzytywnych.

Zasady projektowania

- Jeden obiekt – jedna tablica.
- Atrybuty <> obiekty.
- Atomiczność atrybutów.
- Atrybuty – tylko raz, przy „swoim" obiekcie.
- Encje asocjacyjne – osobne tablice.
- Dekompozycja M:N do 1:N.
- Eliminacja redundancji.

Przykłady

Systemy nietrywialne

- baza danych dla koleji,
- baza danych dla MPK,
- lotnicze bazy danych,
- banki,
- SOPD, Dyplomant,
- baza danych parafii (księgi metrykalne),
- sklep z częściami elektronicznymi,
- telekomunikacja,
- dane uzaleznione przestrzennie i czasowo,
- działki, drogi, mapy.