

Bazy danych



Andrzej Łachwa, UJ, 2015

andrzej.lachwa@uj.edu.pl

9/14

Rozkłady schematów

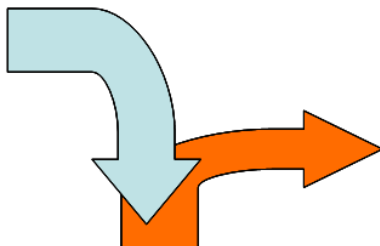
Proces normalizacji polega na „rozbijaniu” schematów relacji na mniejsze, aż przestanie to być możliwe lub pożądane. Rozkłady powinny pozwalać na złączenie nieaddytywne (odwracalność) oraz w miarę możliwości zachowywać zależności.

Rozkład zachowuje zależności jeżeli zbiór zależności w podschematach jest równoważny początkowemu zbiorowi zależności.

Rozkład schematu na podschematy nazywamy odwracalnym wtw gdy każda relacja rozpięta na tym schemacie jest złączeniem naturalnym swoich rzutów na podschematy.

Przykład: $T(S)$, $S=\{A, B, C\}$, $\rho=(T_1\{A, B\}, T_2\{A, C\})$

A	B	C
a	c	e
b	c	f
b	d	e



A	B	C
a	c	e
b	c	f
b	c	e
b	d	f
b	d	e

Nie każdy
rozkład jest
odwracalny!

A	C
a	e
b	f
b	e

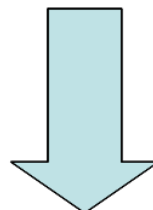
A	B
a	c
b	c
b	d

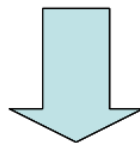
$T(S) = T \{NrFakt, NrPoz, IdTow, Ilość, NrKlienta\}$

$\rho = (T_1 \{NrFakt, NrPoz, IdTow, Ilość\},$

$T_2 \{NrFakt, NrKlienta\})$

NrFakt	NrPoz	IdTow	Ilość	NrKli
123	1	TV11	1	52
123	2	RA76	1	52
123	3	KA3	2	52
124	1	FR5	4	1336
124	2	FR7	4	1336





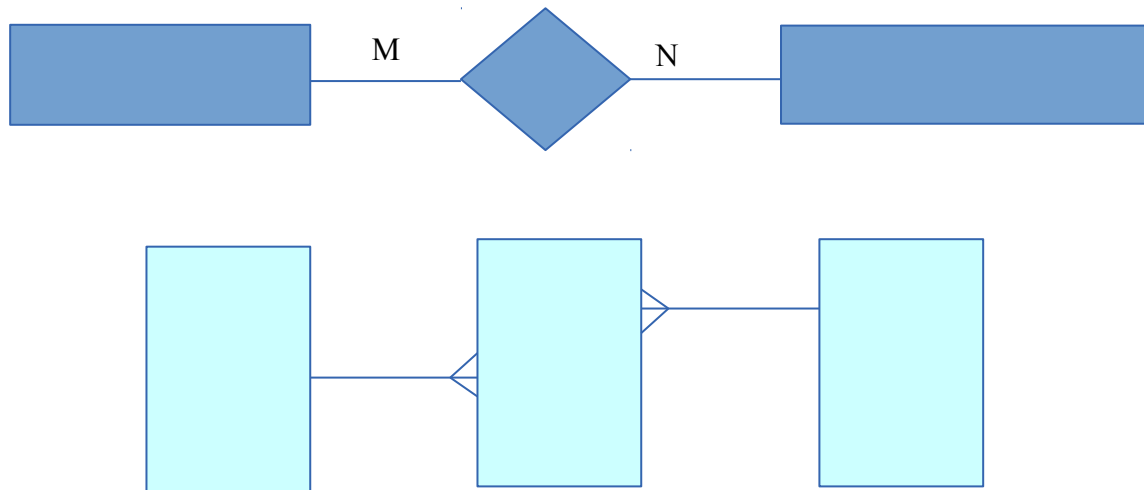
NrFakt	NrPoz	IdTow	Ilość
123	1	TV11	1
123	2	RA76	1
123	3	KA3	2
124	1	FR5	4
124	2	FR7	4

NrFakt	NrKli
123	52
124	1336

**Ten rozkład jest
odwracalny!**

TABELA POŁĄCZEŃ

Rozważmy typowy przykład związku binarnego wiele do wielu oraz odpowiadającego mu schematu złożonego z 3 tabel.



```
CREATE TABLE Chłopcy (ImięCh VARCHAR(30) PRIMARY KEY);  
CREATE TABLE Dziewczyny (ImięD VARCHAR(30) PRIMARY KEY);  
CREATE TABLE Pary (ImięCh VARCHAR(30), ImięD VARCHAR(30));
```

Do tabeli Pary możemy wstawić przykładowe wiersze:

- (Adam, Anna)
- (Adam, Barbara)
- (Karol, Barbara)
- (Adam, Celina)
- (Adam, Anna)

Teraz wprowadzamy więzy integralności. Po pierwsze, nie chcemy dopuszczać wierszy typu (Adam, NULL) czy (NULL, Celina).

```
CREATE TABLE Pary (  
    ImięCh VARCHAR(30) NOT NULL,  
    ImięD VARCHAR(30) NOT NULL);
```

Po drugie, chcemy aby chłopcy i dziewczyny były wybierane z tabel odpowiadającym typom encji Chłopcy i Dziewczyny.

Po trzecie, gdybyśmy chcieli poprawić pisownię imienia w którejś z tabel łączonych, to zmiana ta powinna zostać odzwierciedlona w tabeli połączeń.

Po czwarte, gdybyśmy chcieli usunąć z bazy jakąś osobę (encję), to wszystkie pary zawierające imię tej osoby również powinny być usunięte.


```
CREATE TABLE Pary (  
    ImięCh VARCHAR(30) NOT NULL  
        REFERENCES Chłopcy (ImięCh)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE,  
    ImięD VARCHAR(30) NOT NULL  
        REFERENCES Dziewczyny (ImięD)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE);
```

Po piąte, chcemy zapobiegać powtarzającym się informacjom (w naszym zestawie danych wiersz 1 i wiersz 5).

```
CREATE TABLE Pary (  
    ImięCh VARCHAR(30) NOT NULL  
        REFERENCES Chłopcy (ImięCh)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE,  
    ImięD VARCHAR(30) NOT NULL  
        REFERENCES Dziewczyny (ImięD)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE,  
    PRIMARY KEY (ImięCh, ImięD));
```

Po szóste', chcemy by chłopak mógł być w kilku parach, a każda dziewczyna tylko w jednej!

```
CREATE TABLE Pary (  
    ImięCh VARCHAR(30) NOT NULL  
        REFERENCES Chłopcy (ImięCh)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE,  
    ImięD VARCHAR(30) NOT NULL UNIQUE  
        REFERENCES Dziewczyny (ImięD)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE,  
    PRIMARY KEY (ImięCh, ImięD));
```

Po szóste", chcemy by dziewczyna mogła być w kilku parach, a każda chłopak tylko w jednej! I wreszcie po siódme, ograniczymy się do związków monogamicznych.

```
CREATE TABLE Pary (  
    ImięCh VARCHAR(30) NOT NULL UNIQUE  
    REFERENCES Chłopcy (ImięCh)  
    ON UPDATE CASCADE  
    ON DELETE CASCADE,  
    ImięD VARCHAR(30) NOT NULL UNIQUE  
    REFERENCES Dziewczyny (ImięD)  
    ON UPDATE CASCADE  
    ON DELETE CASCADE,  
    PRIMARY KEY (ImięCh, ImięD));
```

Mamy tu tzw. klucze zagnieźdżone i klucz nadmiarowy.

MODELOWANIE HIERARCHII KLAS

W wielu projektach występują klienci-osoby i klienci-firmy. Wygodnie jest modelować tę sytuację jako związek klasy KLIENCI z podklasami OSOBY i FIRMY.

```
CREATE TABLE Klienci (  
    id CHAR(5) PRIMARY KEY,  
    typ CHAR(1) NOT NULL CHECK(typ IN('O', 'F')),  
    staż TINYINT UNSIGNED NOT NULL DEFAULT 0,  
    obroty DECIMAL(5,2),  
    ...,  
    UNIQUE (id, typ));
```

Jakie ograniczenia wprowadzić w podklasach?

```
CREATE TABLE Osoby (  
    id CHAR(5) PRIMARY KEY,  
    typ CHAR(1) DEFAULT 'O' NOT NULL CHECK (typ='O'),  
    UNIQUE (id, typ)  
    FOREIGN KEY (id, typ)  
    REFERENCES Klienci(id, typ)  
    ON UPDATE CASCADE  
    ON DELETE CASCADE  
    ..... );
```

```
CREATE TABLE Firmy (  
    id CHAR(5) PRIMARY KEY,  
    typ CHAR(1) DEFAULT 'F' NOT NULL CHECK(typ='F'),  
    UNIQUE (id, typ) .....
```

Teraz można to wszystko ukryć w widokach:

```
CREATE VIEW NasiKlienci (Id, Staż, Obroty, ...)  
AS (SELECT Id, Staż, Obroty ... FROM Klienci);
```

```
CREATE VIEW OsobyFizyczne (Id, Tytuł, Imię, Nazwisko ...)  
AS (SELECT Id, Tytuł, Imię, Nazwisko ... FROM Osoby);
```

```
CREATE VIEW OsobyPrawne (Id, Nazwa, FormaPrawna, ...)  
AS (SELECT Id, Nazwa, FormaPrawna, ... FROM Firmy);
```


NORMALIZACJA: zależności wielowartościowe

Zależność taka powstaje wtedy, gdy encje danego typu posiadają dwa niezależne atrybuty wielowartościowe. Np. student zna kilka języków obcych i kilka języków programowania. W odpowiedniej relacji musi wtedy wystąpić wiele krotek dla każdego studenta i muszą być powtarzane wszystkie wartości jednego atrybutu dla każdej wartości drugiego:

Adam Abacki; rosyjski; C++

Adam Abacki; rosyjski; Java

Adam Abacki; rosyjski; SQL

Adam Abacki; angielski; C++

Adam Abacki; angielski; Java

Adam Abacki; angielski; SQL

Zależność wielowartościowa pojawia się nie tylko wtedy, gdy w ramach jednego schematu połączymy dwa atrybuty wielowartościowe jednego typu encji, ale także wtedy gdy w jednym schemacie przedstawimy dwa związki jednego typu encji.

Np. tabelę przedstawioną wyżej można skojarzyć ze strukturą encji typu STUDENT o dwóch atrybutach wielowartościowych: JĘZYK_NATURALNY, JĘZYK_PROGRAMOWANIA. Tę samą strukturę informacji można przedstawić jako dwa związki wychodzące z encji typu STUDENT: związek z encjami JĘZYK_NATURALNY i związek z encjami JĘZYK_PROGRAMOWANIA. Zamiana na schemat uwzględniający oba związki doprowadzi do powstania zależności wielowartościowej.

Zauważmy, że prawidłowo wykonane mapowanie dla obu podanych struktur nie doprowadzi do omawianej tabeli z zależnością wielowartościową.

W przypadku dwóch związków na diagramie będziemy mieć typy encji STUDENT, JĘZYK_P, JĘZYK_O i dwa związki łączące studenta z językami. Oba związki są typu wiele do wielu i powinny przejść w oddzielne tabele!

Definicja

Między zbiorami atrybutów X, Y schematu R zachodzi zależność wielowartościowa $X \twoheadrightarrow Y$ wtw gdy dla każdej relacji r rozpiętej na R zachodzi:

jeżeli w r istnieją krotki t_1 i t_2 takie że $t_1[X]=t_2[X]$

to istnieją również krotki t_3 i t_4 takie że

- $t_1[X]=t_2[X]=t_3[X]=t_4[X]$
- $t_1[Y]=t_3[Y]$ oraz $t_2[Y]=t_4[Y]$
- $t_2[Z]=t_3[Z]$ oraz $t_1[Z]=t_4[Z]$

gdzie $Z=R - (X \cup Y)$.

Kiedy w R zachodzi $X \twoheadrightarrow Y$ to również zachodzi $X \twoheadrightarrow Z$.

Zależność wielowartościową $X \twoheadrightarrow Y$ nazywamy trywialną gdy $Y \subset X$ lub $R = X \cup Y$.

Reguły dla zależności wielowartościowych:

- (1) $X \twoheadrightarrow Y \vdash X \twoheadrightarrow (R - XY)$ (uzupełnianie)
- (2) $X \twoheadrightarrow Y, W \supset Z \vdash WX \twoheadrightarrow YZ$ (zwiększanie)
- (3) $X \twoheadrightarrow Y, Y \twoheadrightarrow Z \vdash X \twoheadrightarrow (Z - Y)$ (przechodność)

gdzie X, Y, Z, W to zbiory atrybutów z $R(A_1, A_2, \dots, A_n)$,

napis PQ oznacza dla atrybutów $\{P, Q\}$ a dla zbiorów atrybutów $P \cup Q$, oraz \vdash jest znakiem inferencji.

Reguły dla zależności funkcyjnych i wielowartościowych:

(4) $X \rightarrow Y \vdash X \twoheadrightarrow Y$ (replikacja)

(5) $X \twoheadrightarrow Y$ oraz istnieje W taki że $W \cap Y = \emptyset$, $W \rightarrow Z$, $Y \supset Z$
 $\vdash X \rightarrow Z$ (scalanie)

Reguły Armstronga, reguły (1)-(3) oraz reguły (4) i (5) stanowią łącznie minimalny i zupełny zbiór reguł wnioskowania dla zależności obu typów (funkcyjnych i wielowartościowych).

Używając tych reguł można zbudować domknięcie dowolnego zbioru zależności obu typów.

4NF

Schemat relacji R ze zbiorem F zależności funkcyjnych i zależności wielowartościowych znajduje się w czwartej postaci normalnej gdy dla każdej nietrywialnej zależności $X \twoheadrightarrow Y$ w domknięciu F^+ zbiór X jest nadkluczem.

$$R(A, B, C, D, E); AB \twoheadrightarrow C$$

wtedy na podstawie (1) mamy $AB \twoheadrightarrow DE$

(obie zależności są nietrywialne)

rozkładamy schemat:

$$R_1(A, B, C); AB \twoheadrightarrow C \quad i \quad R_2(A, B, D, E); AB \twoheadrightarrow DE$$

Zależności łączeniowe i 5NF

Może się zdarzyć, że schemat jest już w 4NF oraz istnieje odwracalny rozkład na trzy lub więcej podschematów. Mówimy wtedy o występowaniu zależności łączeniowej.

Schemat jest w 5NF jeżeli dla każdej nietrywialnej zależności łączeniowej każdy podschemat wynikający z tej zależności zbudowany jest z atrybutów stanowiących nadklucz klucza schematu.

Przykład [Elmasri, Navathe, s.373]

Dostawa (Dostawca, Część, Projekt)

Nowak, śruba, P1

Nowak, nakrętka, P2

Adamski, śruba, P2

Warecki, nakrętka, P3

Adamski, gwóźdź, P1

Adamski, śruba, P1

Nowak, śruba, P2

R1 (Dostawca, Część), R2 (Dostawca, Projekt), R3 (Część, Projekt)

Postać normalna klucza dziedziny DKNF

Jest to ostateczna postać normalna, która uwzględnia wszystkie rodzaje zależności i więzów. Schemat jest w DKNF jeżeli wszystkie więzy i zależności, które powinny być zachowane w każdej relacji rozpiętej na tym schemacie, mogą być wymuszane przez więzy domenowe i więzy kluczy (kluczy głównych i kluczy obcych).

Wymuszenie ogólnych więzów integralności (np., że średnia płaca pracownika danej jednostki nie może być mniejsza od połowy płacy kierownika tej jednostki) przez domeny i klucze może być niewykonalne. Zwykle konieczne jest zdefiniowanie odpowiedniej procedury, asercji czy triggera.

Przegląd typów danych w MySQL

Zob. *MySQL 5.6 Reference Manual*