Normalizacja

- Anomalie wstawiania, usuwania i aktualizacji
- Nieformalne wytyczne dotyczące normalizacji
- Zależności funkcyjne i atrybuty podstawowe
- Postaci normalne

Anomalie wstawiania, usuwania i aktualizacji

SUPPLIES

SUPNR	***PRODNR***	**PURCHASE_PRICE**	**DELIV_PERIOD**	***SUPNAME***	***SUPADDRESS***	**…**	***PRODNAME***	***PRODTYPE***	**…**
21	0289	17.99	1	*Deliwines*	*240, Avenue of the Americas*		*Chateau Saint Estève de Neri, 2015*	*Rose*	
21	0327	56.00	6	*Deliwines*	*240, Avenue of the Americas*		*Chateau La Croix Saint-Michel, 2011*	*Red*	
…									

PO_LINE

PONR	***PRODNR***	**QUANTITY**	***PODATE***	***SUPNR***
1511	0212	2	*2015-03-24*	*37*
1511	0345	4	*2015-03-24*	*37*
…				

Anomalie wstawiania, usuwania i aktualizacji

Supplier

SUPNR	SUPNAME	SUPADDRESS	SUPCITY	SUPSTATUS
21	Deliwines	240, Avenue of the Americas	New York	20
32	Best Wines	660, Market Street	San Francisco	90
...				

Product

PRODNR	PRODNAME	PRODTYPE	AVAILABLE_QUANTITY
0119	Chateau Miraval, Cotes de Provence Rose, 2015	rose	126
0384	Dominio de Pingus, Ribera del Duero, Tempranillo, 2006	red	38
...			

Supplies

SUPNR	PRODNR	PURCHASE_PRICE	DELIV_PERIOD
21	0119	15.99	1
21	0384	55.00	2
...			

Purchase_Order

PONR	PODATE	SUPNR
1511	2015-03-24	37
1512	2015-04-10	94
...		

PO_Line

PONR	PRODNR	QUANTITY
1511	0212	2
1511	0345	4
...		

Cechy dobrych projektów relacyjnych

- Relacja *in_dep*

ID	*name*	*salary*	*dept_name*	*building*	*budget*
22222	Einstein	95000	Physics	Watson	70000
12121	Wu	90000	Finance	Painter	120000
32343	El Said	60000	History	Painter	50000
45565	Katz	75000	Comp. Sci.	Taylor	100000
98345	Kim	80000	Elec. Eng.	Taylor	85000
76766	Crick	72000	Biology	Watson	90000
10101	Srinivasan	65000	Comp. Sci.	Taylor	100000
58583	Califieri	62000	History	Painter	50000
83821	Brandt	92000	Comp. Sci.	Taylor	100000
15151	Mozart	40000	Music	Packard	80000
33456	Gold	87000	Physics	Watson	70000
76543	Singh	80000	Finance	Painter	120000

- Jest powtarzanie informacji
- Konieczność użycia wartości null (jeżeli dodamy nowy wydział bez instruktorów)

Cechy relacji *in_dep*

- Anomalie - problemy powstające w przypadku, gdy chcemy włączyć zbyt dużo informacji do pojedynczej relacji
  - redundancja - informacje niepotrzebnie powielane w kilku krotkach
  - anomalia wprowadzania danych
  - anomalia usuwania danych
  - anomalia aktualizacji danych

ID	*name*	*salary*	*dept_name*	*building*	*budget*
22222	Einstein	95000	Physics	Watson	70000
12121	Wu	90000	Finance	Painter	120000
32343	El Said	60000	History	Painter	50000
45565	Katz	75000	Comp. Sci.	Taylor	100000
98345	Kim	80000	Elec. Eng.	Taylor	85000
76766	Crick	72000	Biology	Watson	90000
10101	Srinivasan	65000	Comp. Sci.	Taylor	100000
58583	Califieri	62000	History	Painter	50000
83821	Brandt	92000	Comp. Sci.	Taylor	100000
15151	Mozart	40000	Music	Packard	80000
33456	Gold	87000	Physics	Watson	70000
76543	Singh	80000	Finance	Painter	120000

Anomalie wstawiania, usuwania i aktualizacji

- Aby mieć dobry relacyjny model danych, wszystkie relacje w modelu powinny być znormalizowane
- Procedura formalnej normalizacji do transformacji modelu relacyjnego nieznormalizowanego w znormalizowany – dekompozycja tabel
- Korzyści:
  - Na poziomie logicznym użytkownicy mogą łatwo zrozumieć znaczenie danych i formułować poprawne zapytania
  - Na poziomie implementacyjnym przestrzeń dyskowa jest efektywnie wykorzystywana i zmniejsza się ryzyko niespójnych aktualizacji

Dekompozycja

- Jedynym sposobem uniknięcia problemu powtarzania się informacji w schemacie jest zdekomponowanie go na dwa schematy
- Nie wszystkie dekompozycje są poprawne, np. dekompozycja

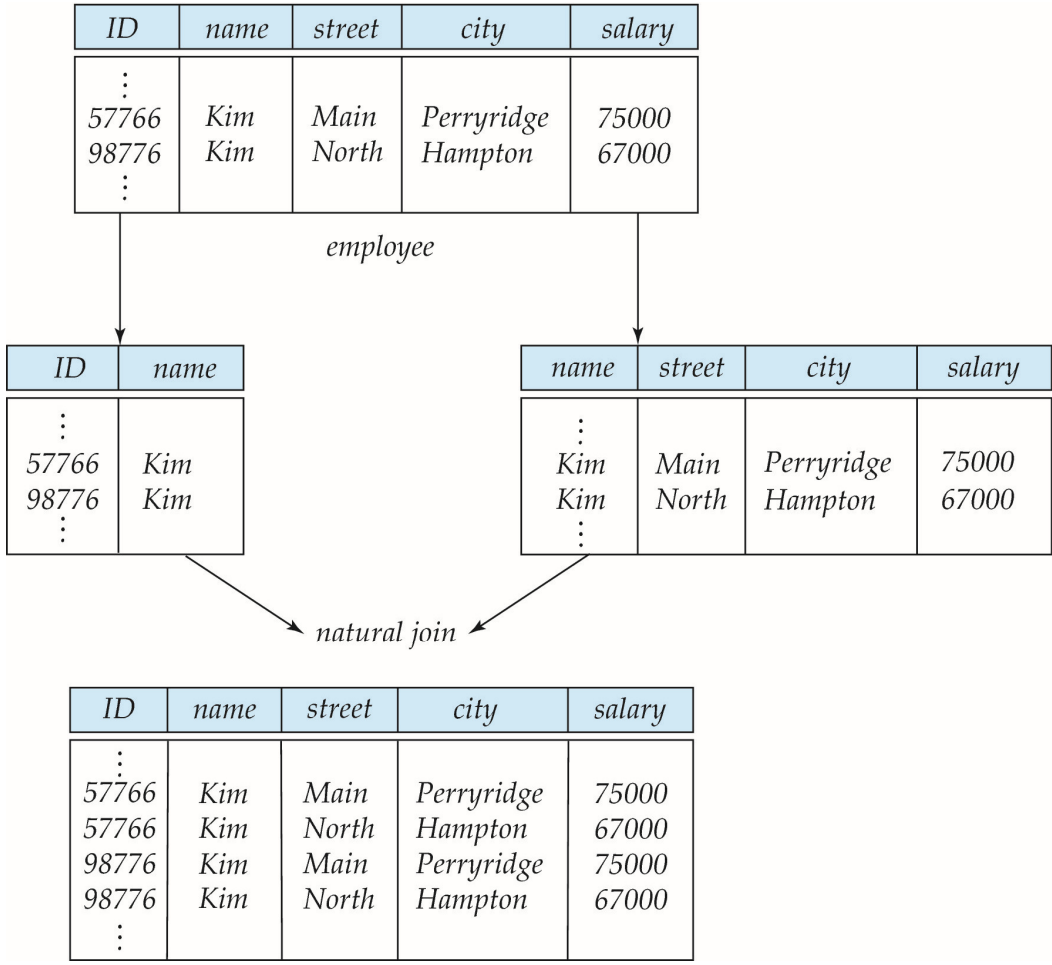
  employee(ID, name, street, city, salary)

  do

  employee1 (*ID*, *name*)

  employee2 (*name*, *street, city, salary*)

  Problem, gdy dwóch pracowników z tym samym nazwiskiem
- Tracimy informację – nie możemy zrekonstruować pierwotnej relacji *employee* -- więc jest to **dekompozycja stratna**.



Dekompozycja bezstratna

- Niech *R* będzie schematem relacji a R_1 i R_2 tworzą rozkład *R*, tzn R = R_1 U R_2
- Mówimy, że **dekompozycja jest bezstratna** jeżeli nie ma utraty informacji poprzez zastąpienie R dwoma schematami relacji R_1 U R_2
- Formalnie,

$$\prod_{R_1}(r) \bowtie \prod_{R_2}(r) = r$$

- I odwrotnie, rozkład jest stratny, jeżeli

$$r \subset \prod_{R_1}(r) \bowtie \prod_{R_2}(r)$$

Przykład bezstratnej dekompozycji

- Dekompozycja *R = (A, B, C)*

$R_1 = (A, B)$ $R_2 = (B, C)$

A	B	C
α	1	A
β	2	B

r

A	B
α	1
β	2

$\Pi_{A,B}(r)$

B	C
1	A
2	B

$\Pi_{B,C}(r)$

$\Pi_A (r) \bowtie \Pi_B (r)$

A	B	C
α	1	A
β	2	B

Normalizacja

- Decyzja, czy konkretna relacja *R* jest w "dobrej" postaci.
- W przypadku gdy relacja *R* nie jest w "dobrej" postaci, dekompozycja do zbioru relacji $\{R_1, R_2, ..., R_n\}$ takich że
  - Każda relacja jest w dobrej postaci
  - Dekompozycja jest bezstratna
- Teoria oparta jest o:
  - zależności funkcyjne
  - zależności wielowartościowe

Zależności funkcyjne

- Zależność funkcyjna X $\rightarrow$ Y, między dwoma zbiorami atrybutów X i Y implikuje, że wartość X jednoznacznie określa wartość Y
  - istnieje zależność funkcyjna od X do Y lub Y jest funkcyjnie zależne od X
- np.:
  - SSN $\rightarrow$ ENAME
  - PNUMBER $\rightarrow$ {PNAME, PLOCATION}
  - {SSN, PNUMBER} $\rightarrow$ HOURS

Definicja zależności funkcyjnych

- Niech *r(R)* będzie schematem relacji

$$\alpha \subseteq R \;\; i \;\; \beta \subseteq R$$

- **Zależność funkcyjna**

$$\alpha \rightarrow \beta$$

zachodzi na *R* wtedy i tylko wtedy gdy dla każdej relacji *r*(R), jeżeli dowolne dwie krotki t_1 *i* t_2 w *r* są zgodne w atrybutach α, są również zgodne w atrybutach β. Czyli,

$$t_1[\alpha] = t_2[\alpha] \;\; \Rightarrow \;\; t_1[\beta] = t_2[\beta]$$

- Np: Niech *r*(A,*B*) z następującą instancją *r.*

1	4
1	5
3	7

- W tej instancji $B \rightarrow A$ zachodzi; a $A \rightarrow B$ **NIE** zachodzi,

Trywialne zależności funkcyjne

- Zależność funkcyjna jest **trywialna** jeżeli jest spełniona przez wszystkie relacje
  - *Np:*
    - *ID, name* $\rightarrow$ *ID*
    - *name* $\rightarrow$ *name*
  - Ogólnie, $\alpha \rightarrow \beta$ jest trywialna jeżeli $\beta \subseteq \alpha$

Domknięcie zbioru zależności funkcyjnych

- Mając zbiór *F* zależności funkcyjnych, są pewne inne zależności funkcyjne, które są logicznie implikowane przez zbiór *F*.
  - Jeżeli $A \rightarrow B$ i $B \rightarrow C$, można wywnioskować, że zachodzi $A \rightarrow C$
  - etc.
- Zbiór **wszystkich** zależności funkcyjnych logicznie wynikający ze zbioru *F* jest **domknięciem** *F*.
- *Domknięcie* *F* oznaczane jako F^+.

Domknięcie zbioru zależności funkcyjnych

- Można obliczyć F^+ , domknięcie F, przez wielokrotne stosowanie **Aksjomatów Armstronga:**
  - **Reguła zwrotności:** if $\beta \subseteq \alpha$, then $\alpha \rightarrow \beta$
  - **Reguła rozszerzalności:** if $\alpha \rightarrow \beta$, then $\gamma\,\alpha \rightarrow \gamma\,\beta$
  - **Reguła przechodniości:** if $\alpha \rightarrow \beta$, and $\beta \rightarrow \gamma$, then $\alpha \rightarrow \gamma$
- Reguły te
  - generują tylko te zależności funkcyjne, które faktycznie zachodzą i
  - generują wszystkie zależności funkcyjne, które zachodzą.

Domknięcie zbioru zależności funkcyjnych c.d.

- Dodatkowe reguły:
  - **Reguła unii**: Jeżeli zachodzi $\alpha \rightarrow \beta$ *i* $\alpha \rightarrow \gamma$, to zachodzi $\alpha \rightarrow \beta\gamma$.
  - **Reguła dekompozycji**: Jeżeli zachodzi $\alpha \rightarrow \beta\gamma$, to zachodzi $\alpha \rightarrow \beta$ *i* $\alpha \rightarrow \gamma$.
  - **Reguła pseudoprzechodniości**: Jeżeli zachodzi $\alpha \rightarrow \beta$ *i* $\gamma\beta \rightarrow \delta$, to zachodzi $\alpha\gamma \rightarrow \delta$**.**
- Powyższe reguły można wywnioskować z aksjomatów Armstrong'a.

Dekompozycja bezstratna a zależności funkcyjne

- Można użyć zależności funkcyjnych aby pokazać, że pewne dekompozycje są bezstratne.
- W przypadku $R = (R_1, R_2)$, wymagamy dla wszystkich możliwych relacji r o schemacie R

$$r = \prod_{R1}(r) \bowtie \prod_{R2}(r)$$

- Dekompozycja R do R_1 i R_2 jest bezstratna, jeżeli przynajmniej jedna z zależności jest w F^+:
  - $R_1 \cap R_2 \rightarrow R_1$
  - $R_1 \cap R_2 \rightarrow R_2$
  - czyli gdy $R_1 \cap R_2$ tworzy nadklucz albo dla R_1 albo dla R_2
- Np. *in_dep* (*ID, name, salary, dept_name, building, budget*) zdekomponowane do :

  instructor (*ID, name, salary, dept_name*)

  department (*dept_name, building, budget*)

  dept_name $\rightarrow$ *dept_name, building, budget*

ID	*name*	*salary*	*dept_name*	*building*	*budget*
22222	Einstein	95000	Physics	Watson	70000
12121	Wu	90000	Finance	Painter	120000
32343	El Said	60000	History	Painter	50000

Zależności funkcyjne i atrybuty podstawowe

- Atrybut podstawowy, to atrybut, który jest częścią klucza kandydującego
- Np.: R1(SSN, PNUMBER, PNAME, HOURS)
  - Atrybuty podstawowe: SSN i PNUMBER
  - Atrybuty nie-podstawowe: PNAME i HOURS

Nieformalne wytyczne dotyczące normalizacji

- Zaprojektuj model relacyjny w taki sposób, aby łatwo było wyjaśnić jego znaczenie
  - MYRELATION123(SUPNR, SUPNAME, SUPTWITTER, PRODNR, PRODNAME, …) versus SUPPLIER(SUPNR, SUPNAME, SUPTWITTER, PRODNR, PRODNAME, ……)
- Atrybuty z wielu typów encji nie powinny być łączone w jedną relację
  - SUPPLIER(SUPNR, SUPNAME, SUPTWITTER, ……)
- Unikaj nadmiernej liczby wartości NULL w relacji
  - SUPPLIER(SUPNR, SUPNAME, …)
  - SUPPLIER-TWITTER(SUPNR, SUPTWITTER)

Postaci normalne

- Pierwsza postać normalna (1 NF)
- Druga postać normalna (2 NF)
- Trzecia postać normalna (3 NF)
- Postać normalna Boyce-Codd’a (BCNF)
- Czwarta postać normalna (4 NF)

Pierwsza postać normalna (1 NF)

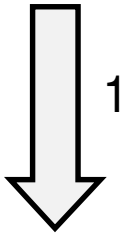
- Mówi, że każdy atrybut relacji musi być niepodzielny i mieć pojedynczą wartość
  - niedopuszczalne atrybuty złożone lub wielowartościowe (ograniczenie dziedziny)
- SUPPLIER(SUPNR, NAME(FIRST NAME, LAST NAME), SUPSTATUS)
- SUPPLIER(SUPNR, FIRST NAME, LAST NAME, SUPSTATUS)

Pierwsza postać normalna (1 NF)

- DEPARTMENT(DNUMBER, DLOCATION, *DMGRSSN*)
  - Założenie: oddział może mieć wiele lokalizacji i wiele oddziałów jest możliwych w danej lokalizacji
- DEPARTMENT(DNUMBER, DMGRSSN)
- DEP-LOCATION(*DNUMBER*, DLOCATION)

Pierwsza postać normalna (1 NF)

DNUMBER	DLOCATION	DMGRSSN
15	{New York, San Francisco}	110
20	Chicago	150
30	{Chicago, Boston}	100



DEPARTMENT

DNUMBER	***DMGRSSN***
15	110
20	150
30	100

DEP-LOCATION

DNUMBER	**DLOCATION**
15	New York
15	San Francisco
20	Chicago
30	Chicago
30	Boston

Pierwsza postać normalna (1 NF)

- R1(SSN, ENAME, DNUMBER, DNAME, PROJECT(PNUMBER, PNAME, HOURS))
  - załóżmy, że pracownik może pracować nad wieloma projektami, a wielu pracowników może pracować nad tym samym projektem
- R11(SSN, ENAME, DNUMBER, DNAME)
- R12(*SSN*, PNUMBER, PNAME, HOURS)

Druga postać normalna (2 NF)

- Zależność funkcyjna X $\rightarrow$ Y jest zupełną zależnością funkcyjną, jeżeli usunięcie dowolnego atrybutu A z X oznacza, że zależność już nie obowiązuje
  - np.: SSN,PNUMBER $\rightarrow$ HOURS; PNUMBER $\rightarrow$ PNAME
- Zależność funkcyjna X $\rightarrow$ Y jest zależnością częściową, jeżeli atrybut A z X można usunąć z X a zależność nadal obowiązuje
  - np.: SSN,PNUMBER $\rightarrow$ PNAME

Druga postać normalna (2 NF)

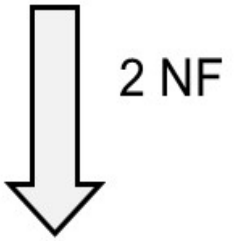
- Relacja R jest w drugiej postaci normalnej (2 NF) jeżeli spełnia 1 NF i każdy atrybut nie-podstawowy A w R jest zupełnie zależy funkcyjnie od dowolnego klucza R
- Jeżeli relacja nie jest w drugiej postaci normalnej należy:
  - zdekomponować ją i stworzyć nową relację dla każdego klucza częściowego wraz z zależnymi atrybutami
  - zostawić relację z oryginalnym kluczem głównym i wszystkim atrybutami, które są od niego zupełnie zależne funkcyjnie

Druga postać normalna (2 NF)

- R1(SSN, PNUMBER, PNAME, HOURS)
  - załóżmy, że pracownik może pracować nad wieloma projektami; nad jednym projektem może pracować wielu pracowników, a projekt ma unikalną nazwę
- R11(SSN, *PNUMBER*, HOURS)
- R12(PNUMBER, PNAME)

Druga postać normalna (2 NF)

SSN	PNUMBER	PNAME	HOURS
100	1000	Hadoop	50
220	1200	CRM	200
280	1000	Hadoop	40
300	1500	Java	100
120	1000	Hadoop	120



PNUMBER	PNAME
1000	Hadoop
1200	CRM
1500	Java

SSN	PNUMBER	HOURS
100	1000	50
220	1200	200
280	1000	40
300	1500	100
120	1000	120

Trzecia postać normalna (3 NF)

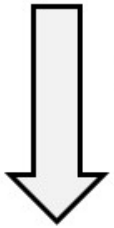
- Zależność funkcyjna $X \rightarrow Y$ w relacji R jest zależnością przechodnią, jeżeli istnieje zbiór atrybutów Z, który nie jest ani kluczem kandydującym, ani podzbiorem żadnego klucza R, i zachodzą zarówno $X \rightarrow Z$ jak i $Z \rightarrow Y$
- Relacja jest w trzeciej postaci normalnej (3 NF) jeżeli spełnia 2 NF i żaden nie-główny atrybut w R nie jest przejściowo zależny od klucza głównego
- Jeśli relacja nie jest w trzeciej postaci normalnej, należy rozłożyć relację R i stworzyć relację, która zawiera atrybuty nie-kluczowe, które funkcyjnie określają inne atrybuty nie-kluczowe

Trzecia postać normalna (3 NF)

- R1(SSN, ENAME, DNUMBER, DNAME, DMGRSSN)
  - Załóżmy, że pracownik pracuje w jednym dziale, dział może mieć wielu pracowników, ale dział ma jednego kierownika
- R11(SSN, ENAME, *DNUMBER*)
- R12(DNUMBER, DNAME, *DMGRSSN*)

Trzecia postać normalna (3 NF)

SSN	NAME	DNUMBER	DNAME	*DMGRSSN*
10	O'Reilly	10	Marketing	210
22	Donovan	30	Logistics	150
28	Bush	10	Marketing	210
30	Jackson	20	Finance	180
12	Thompson	10	Marketing	210



3 NF

SSNR	NAME	*DNUMBER*
10	O'Reilly	10
22	Donovan	30
28	Bush	10
30	Jackson	20
12	Thompson	10

DNUMBER	DNAME	*DMGRSSN*
10	Marketing	210
30	Logistics	150
20	Finance	180

Postać normalna Boyce-Codd'a (BCNF)

- Zależność funkcyjna X → Y jest trywialną zależnością funkcyjną, jeżeli Y jest podzbiorem X
  - np.: SSN, NAME → SSN
- Relacja R jest w BCNF pod warunkiem, że każda z jej nietrywialnych zależności funkcyjnych X → Y, X jest nadkluczem—to znaczy X jest albo kluczem kandydującym albo jego nadzbiorem
- BCNF jest silniejsza niż 3NF
  - każda relacja w BCNF jest również w 3 NF (ale nie odwrotnie)

Postać normalna Boyce-Codd'a (BCNF)

- R1(SUPNR, SUPNAME, PRODNR, QUANTITY)
  - Załóżmy, że dostawca może dostarczyć wiele produktów; produkt może być dostarczany przez wielu dostawców, a dostawca ma unikalną nazwę
- R11(*SUPNR*, PRODNR, QUANTITY)
- R12(SUPNR, SUPNAME)

Czwarta postać normalna (4 NF)

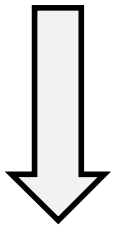
- Istnieje wielowartościowa zależność od X do Y, X →→ Y, wtedy i tylko wtedy, gdy każda wartość X dokładnie określa zbiór wartości Y, niezależnie od innych atrybutów
- Relacja jest w 4 NF, jeżeli jest w BCNF i dla każdej z jej nietrywialnych zależności wielowartościowych X →→ Y, X jest nadkluczem—to znaczy X jest albo kluczem kandydującym lub jego nadzbiorem

Czwarta postać normalna (4 NF)

- R1(course, instructor, textbook)
  - Załóżmy, że kurs może być prowadzony przez różnych instruktorów, a kurs wykorzystuje ten sam zestaw podręczników dla każdego instruktora
- R11(course, textbook)
- R12(course, instructor)

Czwarta postać normalna (4 NF)

COURSE	INSTRUCTOR	BOOK
Database Management	Baesens	Database cookbook
Database Management	Lemahieu	Database cookbook
Database Management	Baesens	Databases for dummies
Database Management	Lemahieu	Databases for dummies



4 NF

COURSE	INSTRUCTOR
Database Management	Baesens
Database Management	Lemahieu

COURSE	BOOK
Database Management	Database cookbook
Database Management	Databases for dummies

Przegląd kroków normalizacji i zależności

Postać normalna	Rodzaj zależności	Opis
2NF	Zupełna zależność funkcyjna	Zależność funkcjonalna X->Y jest zupełną zależnością funkcyjną, jeśli usunięcie dowolnego atrybutu A z X oznacza, że zależność już nie zachodzi
3NF	Przechodnia zależność funkcyjna	Zależność funkcjonalna X->Y w relacji R jest zależnością przechodnią, jeśli istnieje zbiór atrybutów Z, który nie jest ani kluczem kandydującym, ani podzbiorem żadnego klucza R, i zachodzą zarówno X->Z, jak i Z->Y
BCNF	Trywialna zależność funkcyjna	Zależność funkcjonalną X->Y nazywamy trywialną, jeśli Y jest podzbiorem X
4NF	Wielowartościowa zależność	Zależność X->>Y jest wielowartościowa wtedy i tylko wtedy, gdy każda wartość X dokładnie określa zbiór wartości Y, niezależnie od innych atrybutów