Bazy danych

Opracowano na podstawie:
Thomas Connolly, Carolyn Begg, Systemy baz danych, Wydawnictwo RM, 2004

1 Relacyjny model danych

Relacyjny model danych [3] bazuje na matematycznym pojęciu relacji. W tym modelu dane i związki są reprezentowane w postaci tabel, z których każda zawiera pewną liczbę kolumn o unikalnych identyfikatorach.

„Relacja [Encyklopedia PWN] oznacza wszelki związek lub zależność (stosunek) między dwoma (lub więcej) przedmiotami danego rodzaju, pojęciami, wielkościami itp. Relacje mogą być dwu- lub wieloczłonowe. Relacje dwuczłonowe oznacza się zwykle symbolem xRy lub $R(x, y)$ ('x pozostaje w stosunku R do y'), przy czym x i y są zw. członami relacji, czyli relatywami. Relacje wieloczłonowe oznacza się zwykle symbolem $R(x1, x2, ..., xn)$; W teorii mnogości relację (dwuargumentową) definiuje się jako podzbiór kartezjańskiego iloczynu zbiorów $A \times A$ złożonego z tych par $< x, y >$, w których x i y są związane z rozpatrywaną zależnością R; "

„Kartezjański iloczyn zbiorów [Encyklopedia PWN], mat. jedno z pojęć teorii mnogości; kartezjańskim iloczynem zbiorów X i Y nazywa się zbiór wszystkich uporządkowanych par $< x, y >$, w których pierwszy element x należy do zbioru X, a drugi y do zbioru Y; kartezjański iloczyn zbiorów oznacza się symbolem $X \times Y$; analogicznie określa się kartezjański iloczyn zbiorów $X \times Y \times Z$ itd., kartezjański iloczyn zbiorów nazywa się też produktem zbiorów."

2 Logiczne projektowanie bazy danych dla modelu relacyjnego

Logiczne projektowanie bazy danych [3] - proces tworzenia modelu dla informacji wykorzystywanych w przedsiębiorstwie, przystosowanego do konkretnego modelu danych, lecz niezależnego od konkretnego SZBD i innych szczegółów implementacji fizycznej.

Krok 2

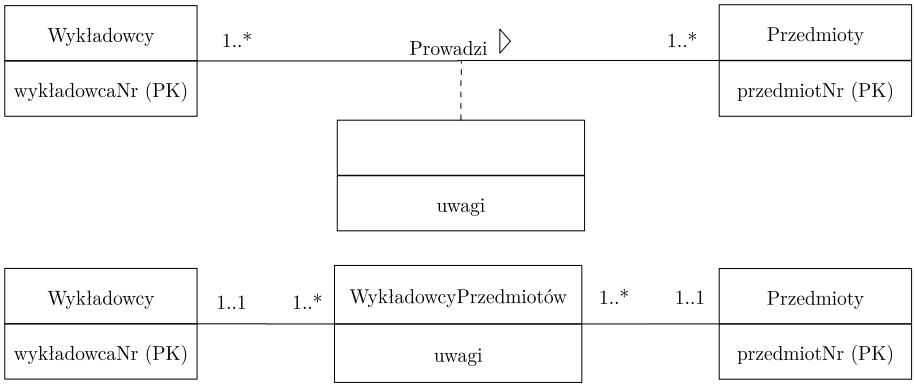
Tworzenie i weryfikacja lokalnego modelu logicznego dla całej perspektywy

Lokalny logiczny model danych składa się z [3]: diagramu ER, schematów relacji oraz dokumentacji.

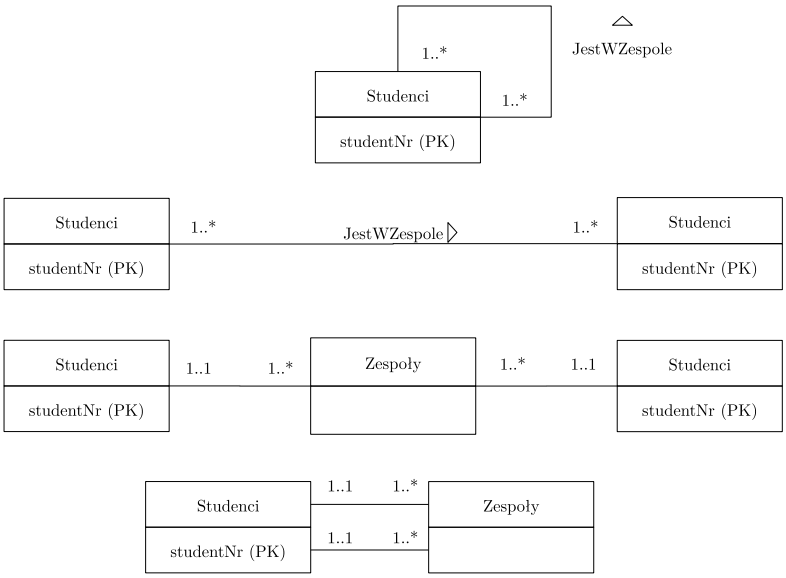
Krok 2.1

Usuń własności niekompatybilne z modelem relacyjnym (krok opcjonalny)

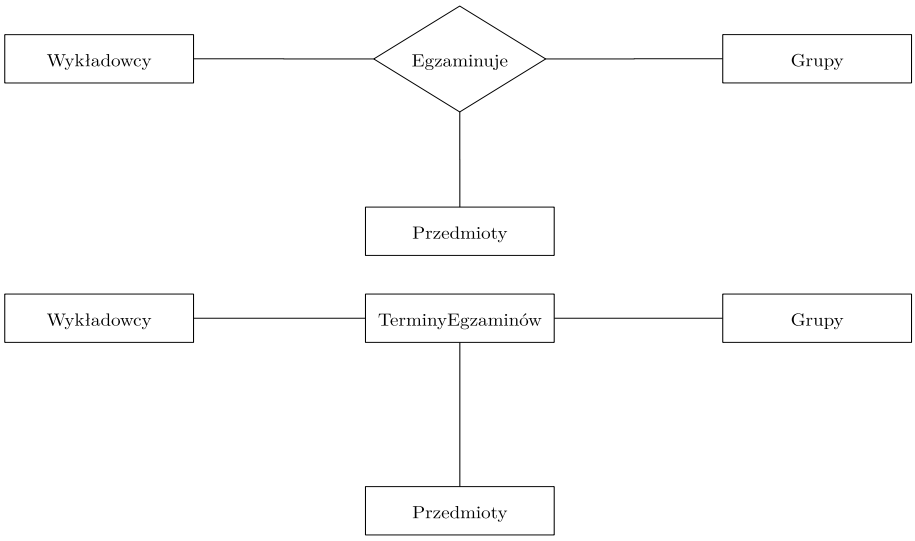
1. Usunięcie binarnych związków typu „wiele do wielu" (*:*).
   Związki binarne wiele do wielu zastępuje się poprzez utworzenie pośredniego zbioru encji oraz zamianę związku „wiele do wielu" na dwa związki typu „jeden do wielu", w których występuje nowy zbiór encji.



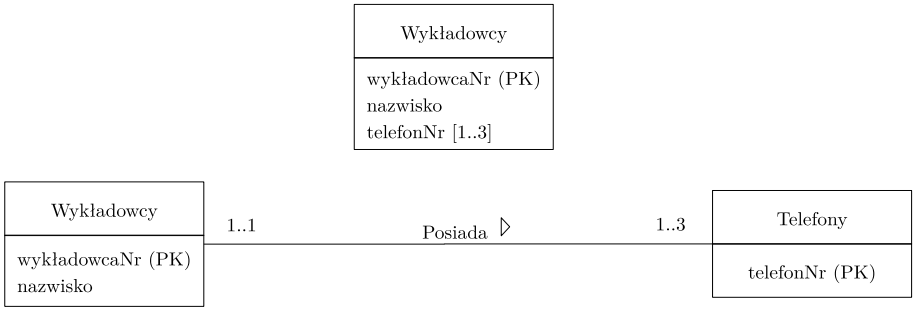
2. Usunięcie rekurencyjnych związków typu „wiele do wielu” (*:*).
   Związki rekurencyjne „wiele do wielu” usuwa się w podobny sposób jak związki binarne typu „wiele do wielu”. Różnica polega ma tym, że w związkach rekurencyjnych ten sam zbiór encji występuje dwa razy (po obu stronach związku).



3. Usunięcie związków złożonych
   W celu usunięcia związku złożonego tworzymy pośredni zbiór encji. Powoduje to, że związek złożony zamieniany jest na odpowiednią liczbę związków binarnych „jeden do wielu” wiążących nowo utworzony zbiór encji ze zbiorami encji występującymi w oryginalnym związku.



4. Usunięcie atrybutów wielowartościowych
   Atrybut wielowartościowy usuwa się poprzez utworzenie nowego zbioru encji.



Informacje o telefonie obejmują: telefon stacjonarny, komórkowy i do pracy.

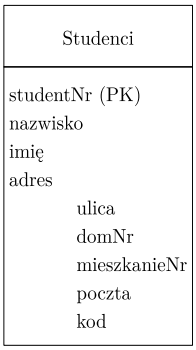
Krok 2.2

Wyznacz relacje dla lokalnego logicznego modelu danych Definicja relacji, która reprezentuje encje, związki i atrybuty w języku definiowania bazy danych (DBDL od ang. Database Defition Language) składa się z nazwy relacji oraz lista prostych atrybutów relacji w nawiasach. W kolejnych wiersza zapisywane są klucz główny, klucze alternatywne i obce, dla których należy podać relację zwierająca wskazany klucz jako klucz główny. Dodatkowo podawana jest lista atrybutów pochodnych oraz sposób ich wyliczania.

Związki pomiędzy encjami reprezentowane są przy pomocy kluczy głównych i kluczy obcych. Relacja podrzędna zawiera klucz obcy, będący kluczem głównym relacji nadrzędnej.

1. Silny zbiór encji

   Dla silnego zbioru encji należy stworzyć relację oraz dodać wszystkie atrybuty proste. Natomiast dla każdego atrybutu złożonego do relacji należy dodać do relacji atrybuty proste, z których się składa.



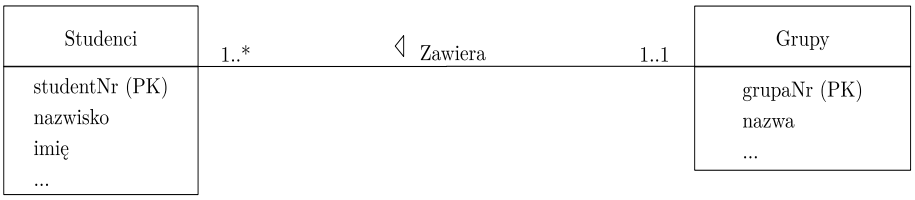

```
Studenci (studentNr, mazwisko, imię, ulica, ..., kod)
Primary Key studentNr;
```

2. Słaby zbiór encji

   Dla słabego zbioru encji należy utworzyć relację zawierającą wszystkie atrybuty proste. Ustalenie klucza głównego jest możliwe dopiero po stworzeniu relacji reprezentujących wszystkie związki z encjami nadrzędnymi ponieważ klucz główny encji słabej jest częściowo lub całkowicie wyznaczany na podstawie encji nadrzędnych.

3. Związek binarny 1:*

   Związek binarny 1:* reprezentujemy poprzez umieszczenie atrybutów klucza głównego encji nadrzędnej (encja po stronie „jeden") w relacji reprezentującej encję podrzędną (encja po stronie „wiele"). Atrybuty te stanowią klucz obcy relacji podrzędnej.




```
Studenci (studentNr, nazwisko, imię, ..., grupaNr)
PrimaryKey studentNr
Foreign Key grupaNr references Grupy(grupaNr);

Grupy(grupaNr, nazwa, ...)
Primary Key grupaNr;
```

4. Binarny związek 1:1

   Sposób reprezentowania związku binarnego 1:1 zależy od rodzaju więzów uczestnictwa:

(a) obowiązkowe po obu stronach:
Należy dwie relacje połączyć, a klucz główny wybrać spośród kluczy głównych połączonych relacji. Pozostały klucz staje kluczem alternatywnym relacji.

(b) obowiązkowe po jednej stronie:
W encji podrzędnej (uczestnictwo obowiązkowe) należy umieścić kopię klucza encji nadrzędnej (uczestnictwo opcjonalne).

(c) opcjonalne po obu stronach:
Jeżeli nie dysponujemy żadnymi dodatkowymi informacjami, to dowolna encja może pełnić funkcję nadrzędną.

5. Związki rekurencyjne 1:1
Podobnie jak dla związków nierekurencyjnych sposób reprezentowania związku zależy od rodzaju więzów uczestnictwa:

(a) obowiązkowe po obu stronach:
Związek reprezentowany jest w postaci jednej relacji z dwoma kopiami klucza głównego, ale jedna z kopi pełni funkcję klucza obcego i posiada nazwę odpowiadająca reprezentowanemu związkowi.

(b) obowiązkowe po jednej stronie:
Istnieją dwie możliwości reprezentowania tego typu związków: w postaci jednej relacji z dwoma kopiami klucza głównego (podobnie jak powyżej) lub poprzez stworzenie nowej relacji reprezentującej związek. Nowa relacja zawiera dwie kopie klucza głównego pełniącymi funkcję klucza obcego i posiadające odpowiednie nazwy.

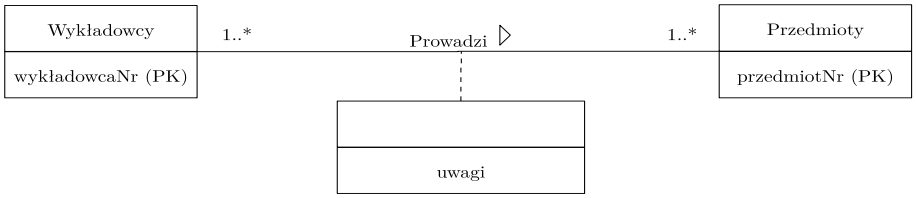
(c) opcjonalne po obu stronach:
Związek reprezentowany jest poprzez stworzenie nowej relacji (podobnie jak powyżej).

6. Związek typu nadklasa-podklasa
Przyjmuje się, że każda nadklasa jest encją nadrzędną, a podklasa encją podrzędną. Istnieje wiele sposobów reprezentacji tego związku.

Więzy uczestnictwa	Więzy rozłączności	Relacje reprezentujące związek
Obowiązkowe	Nierozłączne {I}	Jedna relacja (z jednym lub większą liczbą dyskryminatorów wyznaczających typ każdej krotki).
Opcjonalne	Nierozłączne {I}	Dwie relacje: jedna reprezentująca nadklasę, a druga podklasę (z jednym lub większą liczbą dyskryminatorów wyznaczających typ każdej krotki).
Obowiązkowe	Rozłączne {Lub}	Wiele relacji: osobna relacja dla każdej kombinacji nadklasa - podklasa.
Opcjonalne	Rozłączne {Lub}	Wiele relacji: osobna relacja reprezentująca nadklasę i osobne relacje dla każdej z podklas.

Źródło: T. Connolly, C. Begg, "Systemy baz danych", Wydawnictwo RM, 2004 Więcej informacji można znaleźć w książce: Fowler M., "Architektura systemów zarządzania przedsiębiorstwem. Wzorce projektowe", Helion, 2005

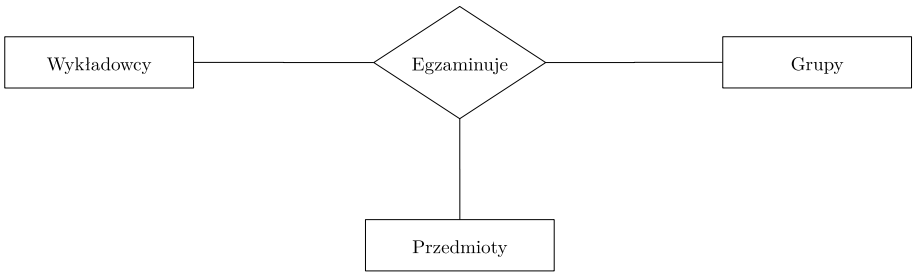
7. Binarne związki typu *:*
Dla każdego związku binarnego wiele do wielu należy utworzyć relację zawierającą atrybuty przypisane do związku oraz kopie atrybutów kluczy głównych wszystkich encji występujących w związku.



```
Wykładowcy (wykładowcaNr, ...)
Primary Key wykładowcaNr;
Przedmioty (przedmiotNr, ...)
Primary Key przedmiotNr;
WykładowcyPrzedmiotów (wykładowcaNr, przedmiotNr, uwagi)
Primary Key wykładowcaNr, przedmiotNr
Foreign Key wykładowcaNr references Wykładowcy(wykładowcaNr)
Foreign Key przedmiotNr references Przedmioty(przedmiotNr);
```

8. Związki złożone

   Dla każdego związku złożonego należy utworzyć relację zawierającą atrybuty przypisane do związku oraz kopie atrybutów kluczy głównych wszystkich encji występujących w związku.

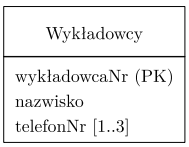



```
TerminyEgzaminów (wykładowcaNr, przedmiotNr, grupaNr)
Primary Key wykładowcaNr, przedmiotNr, grupaNr
Foreign Key wykładowcaNr references Wykładowcy(wykładowcaNr)
Foreign Key grupaNr references Grupy(grupaNr)
Foreign Key przedmiotNr references Przedmioty(przedmiotNr);
```

Założenie: każdy egzamin posiada tylko jeden termin. Dodatkowo istnieją relacje: Wykładowca, Grupa, Przedmiot.

9. Atrybuty wielowartościowe

   Dla każdego atrybutu wielowartościowego należy utworzyć reprezentującą go nową relację oraz umieścić w niej jako klucz obcy klucz główny encji zawierającej ten atrybut.




```
Wykładowcy (wykładowcaNr, ...)
Primary Key wykładowcaNr;
Telefony(telefonID, numer, wykładowcaNr)
Primary Key telefonID
Foreign Key wykładowcaNr references Wykładowcy(wykładowcaNr);
```

Krok 2.3

Wykonaj normalizację relacji

1. Pierwsza postać normalna [2] oznacza po prostu warunek, że każda składowa w każdej krotce ma wartość atomową.

   Pierwsza postać normalna (1NF od ang. first normal form) [3] - to relacja, w której każde przecięcie wiersza i kolumny zawiera jedną i tylko jedną wartość.

   Założenie: czytelnik może pożyczyć daną książkę tylko jeden raz.

id_czytelnika	nazwisko_czytelnika	adres	id_ksiazki	tytul	data_wypozyczenia	data_zwrotu	id_autora	nazwisko_autora
1	Kowalski	Mała 1, 25-900 Kielce	1	Potop	1.01.2001	30.01.2001	1	Sienkiewicz
1	Kowalski	Mała 1, 25-900 Kielce	2	Lalka	2.02.2002	30.04.2002	2	Prus
2	Malinowski	Duża 2, 25-900 Kielce	3	Dziady	3.03.2003	30.06.2003	4	Mickiewicz
2	Malinowski	Duża 2, 25-900 Kielce	4	Kordian	4.04.2004	30.08.2004	3	Słowacki

Większość atrybutów została usunięta w celu poprawienia czytelności.

reader_id	reader_lastname	street	house_numer	...	book_id	title	borrowing_date	returning_date	author_id	author_lastname
1	Kowalski	Mała	1	...	1	Potop	1.01.2001	30.01.2001	1	Sienkiewicz
1	Kowalski	Mała	1	...	2	Lalka	2.02.2002	30.04.2002	2	Prus
2	Malinowski	Duża	2	...	3	Dziady	3.03.2003	30.06.2003	4	Mickiewicz
2	Malinowski	Duża	2	...	4	Kordian	4.04.2004	30.08.2004	3	Słowacki

2. Druga postać normalna (2NF od ang. second normal form) [3] - oznacza relację w pierwszej postaci normalnej, w której każdy atrybut spoza klucza głównego jest od niego w pełni funkcyjnie zależny.

Zależność funkcyjna

Założenia:

- schemat relacyjny zawiera atrybuty (A, B, C, ... Z),
- cała baza jest opisana przez uniwersalną relację $R = (A, B, C, \ldots Z)$.

Zależność funkcyjna opisuje związek pomiędzy atrybutami w relacji. Na przykład, jeśli A i B są atrybutami relacji R, to B jest funkcyjnie zależny od A (oznaczane przez $A \rightarrow B$), jeżeli każda wartość atrybutu A jest powiązana dokładnie z jedną wartością B (A i B mogą się składać z jednego lub większej liczby atrybutów).

Pełna zależność funkcyjna

Można ją zdefiniować następująco: atrybuty B są w pełni zależne funkcyjnie od atrybutów A, jeśli B jest funkcyjnie zależny od A, ale nie jest funkcyjnie zależny od żadnego właściwego podzbioru A.

id_czytelnika	nazwisko_czytelnika	ulica	numer_domu	...
1	Kowalski	Mała	1	...
2	Malinowski	Duża	2	...

3. Trzecia postać normalna (3NF od ang. third normal form) - oznacza relację w pierwszej i drugiej postaci normalnej, w której żaden atrybut spoza klucza głównego nie jest od niego tranzytywnie zależny.

id_ksiazki	tytul	id_autora	nazwisko_autora
1	Potop	1	Sienkiewicz
2	Lalka	2	Prus
3	Dziady	4	Mickiewicz
4	Balladyna	3	Słowacki

id_czytelnika	id_ksiazki	data_wypozyczenia	data_zwrotu
1	1	1.01.2001	30.01.2001
1	2	2.02.2002	30.04.2002
2	3	3.03.2003	30.06.2003
2	4	4.04.2004	30.08.2004

Zależność przechodnia (tranzytywna) występuje w sytuacji, gdy A, B i C są atrybutami relacji, takim że $A \rightarrow B$ i $B \rightarrow C$. Wtedy C jest tranzytywnie zależne od A poprzez B (pod warunkiem, że A nie jest zależne funkcyjnie od B lub C).

id_autora	nazwisko_autora
1	Sienkiewicz
2	Prus
4	Mickiewicz
3	Słowacki

id_czytelnika	nazwisko_czytelnika	ulica	numer_domu	...
1	Kowalski	Mała	1	...
2	Malinowski	Duża	2	...

id_ksiazki	tytul	id_autora
1	Potop	1
2	Lalka	2
3	Dziady	4
4	Balladyna	3

id_czytelnika	id_ksiazki	data_wypozyczenia	data_zwrotu
1	1	1.01.2001	30.01.2001
1	2	2.02.2002	30.04.2002
2	3	3.03.2003	30.06.2003
2	4	4.04.2004	30.08.2004

4. Postać normalna Boyce'a - Codda (BCNF od ang. Boyce-Codd normal form) - „oznacza relację, w której każdy wyznacznik zależności jest kluczem kandydującym"

   Wyznacznik [3] - "określa atrybut lub grupę atrybutów, które występują po lewej stronie strzałki oznaczającej zależność funkcyjną."

   Niezgodność z BCNF występuje dość rzadko gdy np. :

   - relacja zawiera dwa (lub więcej) złożone klucze kandydujące,
   - klucze kandydujące relacji pokrywają się tzn. mają co najmniej jeden wspólny atrybut.

5. Czwarta postać normalna (4NF od ang. fourth normal form),
6. Piąta postać normalna (4NF od ang. fifth normal form),
7. Postać normalna klucza dziedziny (DKNF od ang. domain key normal form),
8. Szósta postać normalna (6NF od ang. sixth normal form).

"Dzięki normalizacji [3] model lepiej odwzorowuje informacje modelowanego przedsiębiorstwa, staje się spójny, zapewnia minimum redundancji i maksimum stabilności. (...) Twierdzi się czasem, że znormalizowana baza danych nie zapewnia maksimum wydajności."

"Relacje zawierające redundantne dane [3] mogą być przyczyną problemów nazywanych anomaliami aktualizacji, które dzielą się na anomalie wstawiania, usuwania i modyfikacji"

Anomalie wstawiania:

- podczas wstawiania studenta trzeba podawać wszystkie informacje o grupie, zgodnie z wartościami z innych wierszy, co może prowadzić do niespójności,
- wstawienie grupy, która nie zawiera studenta wymaga wstawienia wartości pustych do atrybutów dotyczących studenta.

GID	GROUP_NAME	SID	LNAME	FNAME
1	101	7	Kowal	Jan
2	201	3	Malinowski	Henryk
2	202	2	Nowak	Jan
3	301	5	Smith	John
3	301	6	Kowalski	Adam
3	301	4	Polak	Adam
4	401	???	???	???

Anomalie usuwania:

- podczas usunięcia ostatniego studenta należącego do danej grupy tracone są informacje o tej grupie.

GID	GROUP_NAME	SID	LNAME	FNAME
1	101	7	Kowal	Jan
2	201	3	Malinowski	Henryk
2	202	2	Nowak	Jan
3	301	5	Smith	John
3	301	6	Kowalski	Adam
3	301	4	Polak	Adam
4	401	???	???	???

Anomalie modyfikacji:

- podczas zmiany nazwy grupy należy wykonać tę modyfikację w wielu wierszach, co może prowadzić do niespójności.

GID	GROUP_NAME	SID	LNAME	FNAME
1	101	7	Kowal	Jan
2	201	3	Malinowski	Henryk
2	202	2	Nowak	Jan
3	301	5	Smith	John
3	301	6	Kowalski	Adam
3	301	4	Polak	Adam
4	401	???	???	???

Brak anomalii:

SID	LNAME	FNAME
2	Nowak	Jan
3	Malinowski	Henryk
4	Polak	Adam
5	Smith	John
6	Kowalski	Adam

```
GID    GNAME
------ --------------------
     2 201
     3 301
     4 401
```

Krok 2.4

Sprawdź, czy relacje umożliwiają realizację transakcji

Należy sprawdzić czy lokalny logiczny model danych umożliwia wykonanie wszystkich wymaganych transakcji (podobnie jak w kroku 1.8 dla konceptualnego modelu danych).

Krok 2.5

Wyznacz więzy integralności

Więzy integralności to dodatkowe warunki, których spełnienie zapobiega niespójności bazy danych.

1. Wymagana obecność danych (NOT NULL)
2. Więzy dziedzin atrybutów (ustalane wraz z wyborem dziedzin atrybutów w kroku 1.4)
3. Integralność encji - klucz główny nie może przyjmować wartości pustych i oczywiście musi być unikalny dla każdej encji.
4. Integralność referencyjna - oznacza, że jeśli wartość klucza obcego jest określona to musi odpowiadać istniejącej krotce w relacji nadrzędnej. Należy zastanowić się nad następującymi zagadnieniami: Czy dozwolone są wartości puste klucza obcego? Określić warunki, które muszą być spełnione podczas wstawiania, modyfikacji i usuwania klucza kandydującego lub obcego (NO ACTION, CASCADE, SET NULL, SET DEFAULT, NO CHECK).
   Students (studentID, lastname, firstname, ..., groupID)
   PrimaryKey studentID
   Foreign Key groupID references Groups(groupID) ON UPDATE CASCADE, ON DELETE SET NULL;
5. Więzy ogólne - zazwyczaj wynikają z zasad występujących w rzeczywistym świecie np. grupa dziekańska może zawierać od 12 do 32 studentów.

Krok 2.6

Omów lokalny model logiczny z użytkownikami

Należy przeanalizować wspólnie z użytkownikiem model oraz dokumentację.

Krok 3

Tworzenie i weryfikacja globalnego, logicznego modelu danych

Krok 3.1

Scal lokalne, logiczne modele danych w model globalny

Krok 3.2

Sprawdź poprawność globalnego logicznego modelu danych

Krok 3.3

Sprawdź możliwość przystosowania modelu do przewidywanych zmian

Krok 3.4

Omów globalny logiczny model danych z użytkownikami

Literatura

[1] Jeffrey D. Ullman, Jennifer Widom, "Podstawowy wykład z systemów baz danych", WNT, 2001

[2] Jeffrey D. Ullman, Jennifer Widom, "Systemy baz danych, Pełny wykład", WNT, 2006

[3] Thomas Connolly, Carolyn Begg, "Systemy baz danych", Wydawnictwo RM, 2004

[4] Thomas Connolly, Carolyn Begg, "Database systems", Peason, 2015

[5] Kevin Loney, Bob Bryla, Oracle Database 10g, Podręcznik administratora baz danych. Helion 2008

[6] Scott Urman, Ron Hardman, Michael McLaughlin, "Oracle Database 10g, Programowanie w języku PL/SQL", Helion, 2008

[7] Price J, "Oracle Database 11g i SQL Programowanie", Helion, 2009

[8] Ramez Elmasri, Shamkant B. Navathe, "Wprowadzeni do systemów baz danych", Helion, 2005

[9] Sharon Allen, "Modelowanie danych", Helion, 2006

[10] Jose A. Ramalho, Oracle 8i, Mikom, 2001

[11] Dokumentacja do bazy danych Oracle dostępna na stronie www.oracle.com

[12] Sadalage P. J., Fowler M., "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence", Addison-Wesley Professional, 2012

[13] Michael Stonebraker, "New Sql: An Alternative to Nosql and Old Sql For New Oltp Apps", June 16, 2011

[14] Oracle Database PL/SQL Language Reference 11g Release 1 (11.1) B28370-02 (B28370.pdf)

Materiały do przedmiotu dostępne są na stronie: http://weaii-moodle.tu.kielce.pl/