Instytut Informatyki
Wydział Elektrotechniki, Automatyki i Informatyki
Politechnika Opolska

TRANSFORMACJA MODELU ER DO MODELU RELACYJNEGO

Opracowanie: dr inż. Ewelina Piotrowska
e.piotrowska@po.edu.pl
www.e.piotrowska.po.edu.pl

OPOLE 2019

Przedmiot: Modelowanie baz danych

Wprowadzenie

2

- Celem wykładu jest omówienie technik transformacji modelu związków-encji do modelu relacyjnego.
  - Transformacja modelu ER jest konieczna, ponieważ jak pamiętamy jest on modelem abstrakcyjnym niezależnym od implementacji.
  - Modelem relacyjny jest modelem implementacyjnym baz danych. Transformacji podlegają wszystkie obiekty modelu ER, czyli encje z atrybutami, związki i hierarchie encji.
- W ramach wykładu zostaną omówione podstawowe techniki transformacji encji do modelu relacyjnego, transformacji związków i transformacji hierarchii encji.

Pojęcia podstawowe

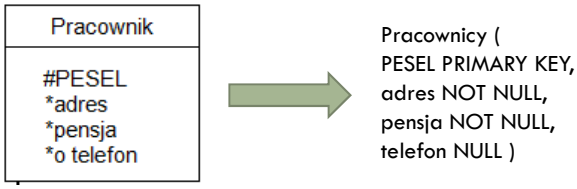
3

- Schemat bazy danych - zbiór schematów relacji
- Relacja (tabela) - dwu-wymiarowa tablica
  - kolumny -> atrybuty
  - wiersze -> krotki, rekordy
    - każda krotka reprezentuje wystąpienie encji
- Klucz podstawowy – atrybut lub zbiór atrybutów - wybrany spośród kluczy potencjalnych
- Klucz obcy – atrybut lub zbiór atrybutów wskazujący na klucz podstawowy innej relacji (atrybut lub zbiór atrybutów w relacji B, będący jednocześnie kluczem podstawowym w relacji A)
  - należy zaznaczyć, że klucz obcy może odnosić się do klucza podstawowego samej relacji, w której został on zdefiniowany

Reguły transformacji encji

4

Model ER	Model relacyjny
Encja	relacja
Atrybut encji	atrybut relacji
Typ danych atrybutu encji	typ danych atrybutu relacji
Identyfikator encji	klucz podstawowy relacji
Obowiązkowość atrybutów encji	ograniczenie NOT NULL
Opcjonalność atrybutów encji	ograniczenie NULL
Pozostałe ograniczenia integr. atrybutów encji	ograniczenia integr. atrybutów relacji



Pracownicy (
PESEL PRIMARY KEY,
adres NOT NULL,
pensja NOT NULL,
telefon NULL)

Reguły transformacji związków

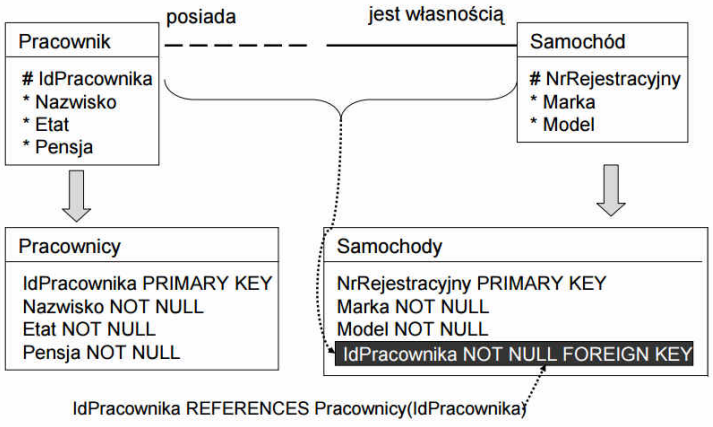
5

Model ER	Model relacyjny
Związek binarny 1:1	klucz obcy we wskazanej tabeli
Związek unarny 1:1	klucz obcy w tej samej tabeli
Związek binarny 1:M	klucz obcy w tabeli po stronie "wiele"
Związek binarny M:N	tabela
Związek unarny M:N	tabela

Związek binarny 1:1

6

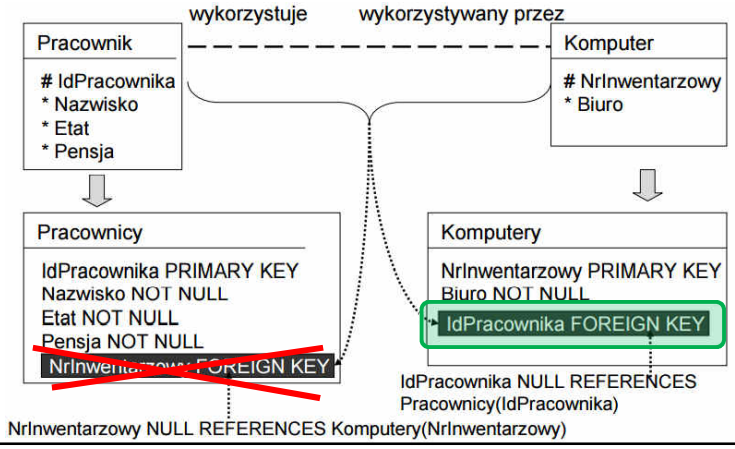
- Związek binarny 1:1 jednostronnie obowiązkowy transformuje się do klucza obcego w tabeli po stronie związku obowiązkowego.
- Ograniczenie integralnościowe jest definiowane dla atrybutu klucza obcego.
- Klucz ten nie może przyjmować wartości pustych.



Związek binarny 1:1

7

- Związek binarny 1:1 obustronnie opcjonalny transformuje się do klucza obcego w tabeli o mniejszym rozmiarze.
- Ograniczenie integralnościowe jest definiowane dla atrybutu klucza obcego.
- Atrybut ten może przyjmować wartości puste, ponieważ związek jest opcjonalny.



Uwagi

8

Przypadek 1 stosowany niezwykle rzadko. Polega on na umieszczeniu kluczy obcych w obu tabelach wynikowych.

1.

2.

Przypadek 2 został omówiony na poprzednim slajdzie

3.

Przypadek trzeci polega na umieszczeniu klucza obcego w tabeli Pracownicy

Związek binarny 1:M

9

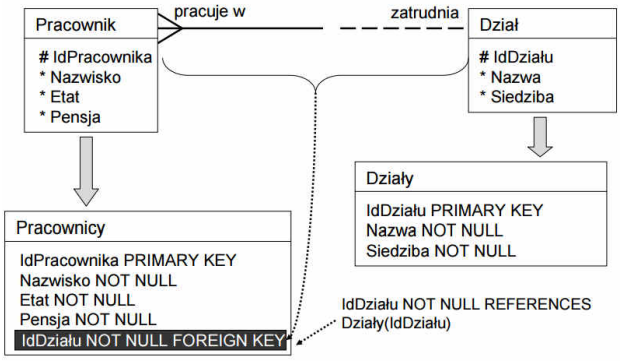
- **Klucz obcy jest dodawany do relacji po stronie "wiele"** niezależnie od opcjonalności, czy obowiązkowości tego związku.

1. **Ograniczenia** integralnościowe referencyjne (tj. definiujące klucz obcy) są definiowane **dla atrybutu reprezentującego klucz obcy**.
2. **Obowiązkowość** związku po stronie "**wiele**" jest reprezentowana przez ograniczenie integralnościowe **NOT NULL** definiowane na kluczu obcym relacji.
3. **Opcjonalność** związku po stronie **"wiele"** jest reprezentowana przez ograniczenie integralnościowe **NULL** definiowaną na kluczu obcym relacji.
4. Opcjonalność lub obowiązkowość związku po stronie "jeden" nie jest odwzorowywana w modelu relacyjnym.

Przykład (związek 1:M)

10

- Przykład sposobu transformacji binarnego związku 1:M jednostronnie obowiązkowego.
- Klucz obcy jest dodawany do tabeli Pracownicy (strona "wiele") i wskazuje on na klucz podstawowy tabeli Działy, czyli IdDziału.
- Klucz obcy IdDziału posiada ograniczenie NOT NULL ponieważ związek jest obowiązkowy od strony "wiele".



Związek binarny M:N

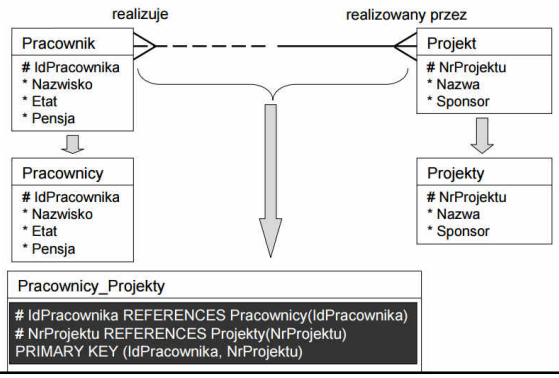
11

- Reguły transformacji związku M:N są identyczne zarówno dla związków jednostronnie obowiązkowych, jak i obustronnie opcjonalnych.
- Związek M:N jest reprezentowany w modelu relacyjnym poprzez dodatkową relację.
- Nazwa relacji reprezentującej związek M:N jest złączeniem nazw relacji powstałych z encji połączonych tym związkiem.
- Relacja dodatkowa zawiera klucze obce wskazujące na klucze podstawowe relacji powstałych z powiązanych encji.
- Ograniczenia referencyjne są definiowane dla kluczy obcych.
- Klucze obce tworzą klucz podstawowy relacji. W konsekwencji, ich wartości nigdy nie będą puste.

Przykład

12

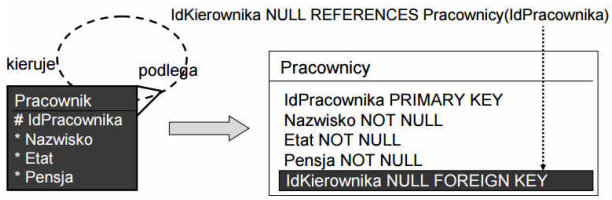
- Sposób transformacji binarnego związku M:N jednostronnie obowiązkowego.
- Ze związku powstaje tabela pośrednia o nazwie Pracownicy_Projekty.
  - Zawiera ona dwa klucze obce.
  - Jeden wskazuje na klucz podstawowy tabeli Pracownicy, czyli IdPracownika, a drugi - na klucz podstawowy tabeli Projekty, czyli NrProjektu.
  - Oba klucze obce stanowią klucz podstawowy tabeli Pracownicy_Projekty.
    - Oznacza to, że ich wartości nigdy nie mogą być puste.



Związek unarny

13

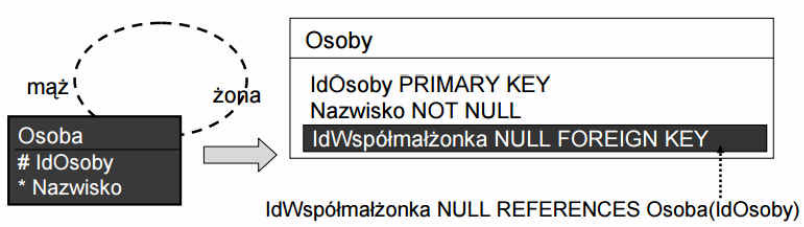
- Związek unarny 1:1 lub 1:M obustronnie opcjonalny transformuje się do klucza obcego w tej samej tabeli.
- Z encji Pracownik powstaje tabela Pracownicy.
  - Zawiera ona klucz obcy IdKierownika wskazujący na IdPracownika w tej samej tabeli.
  - Należy zwrócić uwagę, że wartość klucza obcego może być pusta ponieważ związek jest opcjonalny od strony "wiele".



Przykład – związek unarny

14

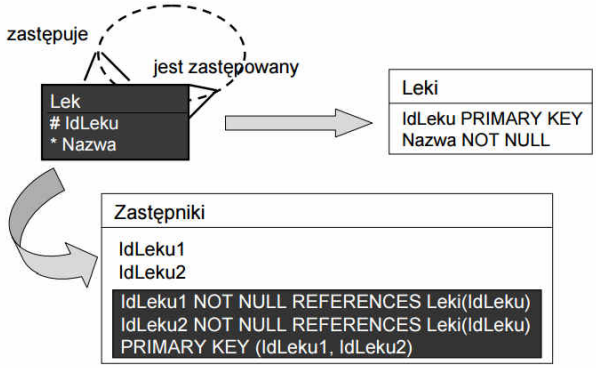
- W tabeli Osoby powstaje klucz obcy IdWspółmałżonka wskazujący na IdOsoby w tej samej tabeli.
  - Ponieważ związek jest opcjonalny, więc klucz obcy może przyjmować wartości puste.



Przykład

15

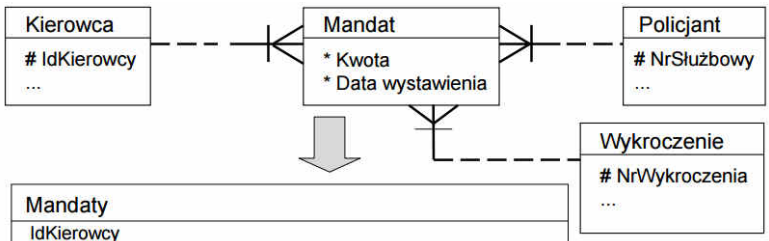
- Związek unarny M:N obustronnie opcjonalny jest transformowany do tabeli pośredniej.
- Z encji Lek powstaje tabela Leki, a związek jest transformowany do tabeli o nazwie Zastępniki.
- Tabela ta posiada dwa klucze obce (atrybut IdLeku1 i IdLeku2), oba wskazują na klucz podstawowy tabeli Leki, czyli na atrybut IdLeku.
- Oba klucze obce wchodzą w skład klucza podstawowego tabeli Zastępniki.

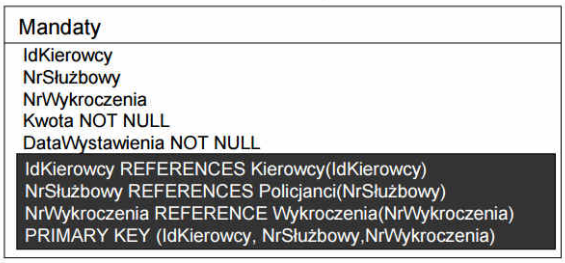


Związki ternarne

16

- Związek ternarny transformuje się w sposób identyczny jak związek 1:M.
- Z encji Mandat powstaje tabela Mandaty.
- Zawiera ona 3 klucze obce: IdKierowcy, NrSłużbowy, NrWykroczenia wskazujące odpwiednio na klucze główne tabel występujących w związku.





Te trzy klucze obce wchodzą w skład klucza podstawowego tabeli Mandaty, ponieważ transformowana encja Mandat była słaba.

Hierarchia encji

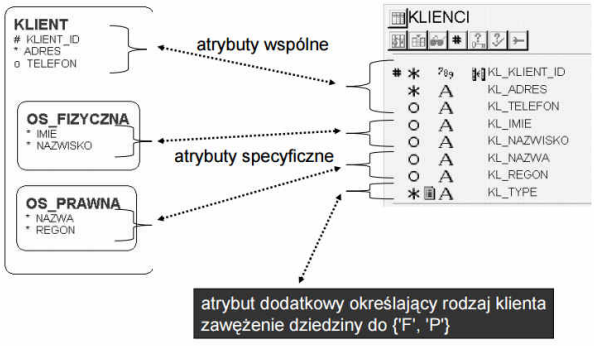
17

- Hierarchię encji do modelu relacyjnego można przetransformować na 3 sposoby.
  - Schemat 1: jedna wspólna tabela ze wszystkimi atrybutami i kluczami obcymi, tj. wspólnymi i specyficznymi dla podencji
  - Schemat 2:
    - dla każdej podencji tworzona osobna tabela ;
    - Każda z tabel zawiera atrybuty wspólne i specyficzne dla określonej encji.
  - Schemat 3:
    - dla atrybutów wspólnych tworzona tabela wspólna
    - dla każdej podencji tworzona osobna tabela z kluczem i atrybutami specyficznymi
    - tabela wspólna i tabele powstałe z podencji powiązane ograniczeniami referencyjnymi

Transformacja hierarchii generalizacji - schemat 1

18

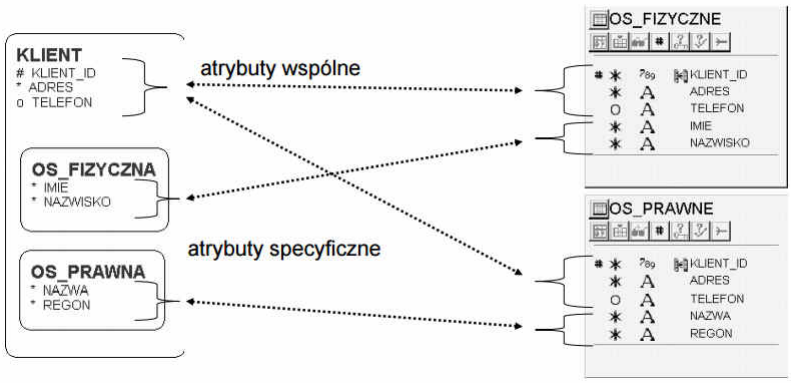
- Powstała tabela Klienci zawiera wszystkie atrybuty wspólne i wszystkie atrybuty specyficzne dla wszystkich podencji.
- Atrybuty specyficzne w tabeli mogą przyjmować wartości puste zawsze, niezależnie od ich definicji w pod-encjach.
- rekord w tabeli klienci opisuje albo klienta fizycznego albo prawnego.
  - Jeśli rekord opisuje klienta fizycznego, to atrybuty klienta prawnego pozostają puste i odwrotnie.
  - Dodatkowo, w tabeli Klienci jest tworzony atrybut z wartością obowiązkową (KL_TYPE), którego wartościami mogą być albo 'F' albo 'P'.
  - Atrybut ten jest przydatny przy wyszukiwaniu klientów określonego typu.



Transformacja hierarchii generalizacji - schemat 2

19

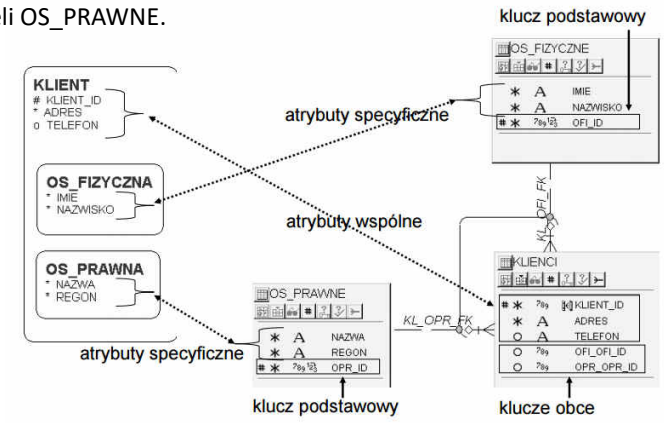
- Z encji OS_FIZYCZNA powstaje tabela OS_FIZYCZNE.
  - Zawiera ona wszystkie atrybuty wspólne encji Klient i wszystkie atrybuty specyficzne encji OS_FIZYCZNA.
  - Opcjonalność/obowiązkowość wartości atrybutów encji przenosi się bezpośrednio na opcjonalność/obowiązkowość atrybutów tabeli OS_FIZYCZNE.
- W podobny sposób jest transformowana encja OS_PRAWNA.



Transformacja hierarchii generalizacji - schemat 3

20

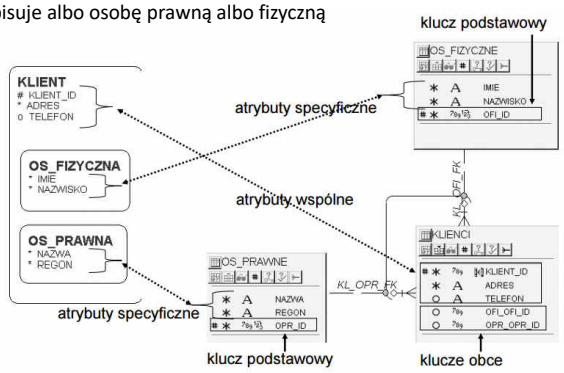
- Z encji OS_FIZYCZNA powstaje tabela OS_FIZYCZNE, która zawiera wszystkie atrybuty specyficzne ze swojej encji i atrybut OFI_ID, który jest kluczem podstawowym tabeli OS_FIZYCZNE.
- Z encji OS_PRAWNA powstaje tabela OS_PRAWNE, która zawiera wszystkie atrybuty specyficzne ze swojej encji i atrybut OPR_ID, który jest kluczem podstawowym tabeli OS_PRAWNE.



CD.

21

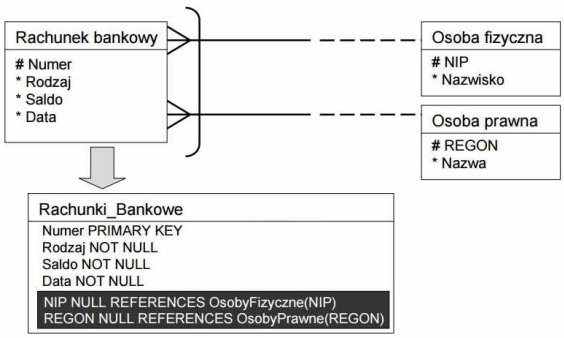
- Z atrybutów wspólnych encji Klient powstaje tabela KLIENCI.
- Dodatkowo, tabela ta posiada dwa klucze obce OFI_OFI_ID i OPR_OPR_ID, z wartościami opcjonalnymi.
  - Pierwszy z nich wskazuje na klucz podstawowy tabeli OS_FIZYCZNE, a drugi - na klucz podstawowy tabeli OS_PRAWNE.
  - Dla danego rekordu w tabeli KLIENCI, tylko jeden klucz obcy może przyjąć wartość, ponieważ rekord w tabeli KLIENCI opisuje albo osobę prawną albo fizyczną



Transformacja związków wyłącznych - schemat 1

22

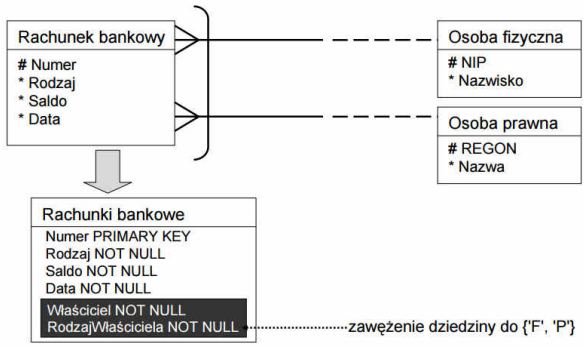
- Transformacja związków wyłącznych jest podobna do transformacji związku 1:M.
  - Z tą tylko różnicą, że klucze obce mogą przyjmować wartości puste.
- Z encji Rachunek bankowy powstaje tabela Rachunki_Bankowe, która posiada dwa klucze obce, jeden wskazuje na klucz podstawowy tabeli Osoby_Fizyczne, a drugi - na klucz podstawowy tabeli Osoby_Prawne.
- Oba klucze obce mogą przyjmować wartości puste, pomimo, że związki z których powstały są obowiązkowe od strony "wiele".
- Dzieje się tak dla tego, że dany rekord rachunku bankowego jest albo związany z osobą fizyczną albo z osobą prawną, więc tylko jeden klucz obcy przyjmie wartość.



Transformacja związków wyłącznych - schemat 2

23

- Alternatywny sposób różni się on od poprzedniego tym, że w tabeli Rachunki_Bankowe jest atrybut Właściciel, który może przyjąć wartość klucza podstawowego rekordu albo z tabeli Osoby_Ficzyzne albo z tabeli Osoby_Prawne;
  - dla atrybutu tego zdefiniowanie ograniczenia referencyjnego jest niemożliwe.
- Ponadto, w tabeli Rachunki_Bankowe jest atrybut RodzajWłaściciela, który może przyjąć jedną z dwóch wartości 'F' lub 'P'. Służy on do określania rodzaju osoby, na którą wskazuje wartość atrybutu Właściciel. Oba atrybuty nie mogą przyjmować wartości pustych.



Etapy projektowania bazy danych

24

Formułowanie definicji celu i założeń wstępnych	Definicja celu określa przeznaczenie bazy danych, a założenia wstępne definiują zadania, które wykonują użytkownicy, wykorzystując dane przechowywane w bazie.
Analiza istniejącej bazy danych	Należy zidentyfikować wymagania organizacji dotyczące danych poprzez analizę sposobu gromadzenia i prezentowania istniejących danych, a także przeprowadzanie rozmów z użytkownikami oraz kierownictwem na temat sposobu codziennego wykorzystywania przez nich tych danych
Tworzenie struktur danych	Tabele tworzone są poprzez identyfikację tematów obsługiwanych przez bazę. Następnie należy skojarzyć każdą tabelę z polami reprezentującymi jej cechę charakterystyczną, a także nadać konkretnemu polu (lub grupie pól) status klucza głównego. Na koniec trzeba określić specyfikacje dla każdego z pól w tabeli.
Określanie i ustalanie zależności w tabelach	Należy zidentyfikować zależności istniejące pomiędzy tabelami w bazie oraz ustalić logiczne powiązania między nimi za pomocą kluczy głównych i obcych lub tabel łączących. Trzeba też opisać każdą zależność, podając jej cechy.
Określanie reguł biznesowych	Trzeba przeprowadzić rozmowy z użytkownikami i kierownictwem w celu określenia ograniczeń, które muszą być nałożone na dane w bazie. Sposób, w jaki organizacja przegląda i wykorzystuje dane, ma wpływ na typ ograniczeń narzucanych bazie. Ograniczenia te nazywane są regułami biznesowymi i pomagają w ustalaniu różnych poziomów integralności danych.
Definiowanie widoków	Rozmowy z pracownikami mają na celu ustalenie różnych sposobów pracy z danymi przechowywanymi w bazie. Po przeprowadzeniu rozmów należy ustalić odpowiednie widoki. Każdy widok definiowany jest przy użyciu odpowiednich tabel i pól. Należy również ustalić kryteria dla tych widoków ograniczające ilość rekordów.
Kontrola integralności danych	Ta faza składa się z czterech etapów. Przede wszystkim należy dokonać analizy każdej tabeli, by upewnić się, że spełnia ona odpowiednie kryteria projektowe. Następnie trzeba sprawdzić specyfikacje wszystkich pól. Kolejnym krokiem jest przetestowanie poprawności wszystkich zależność. Na końcu należy ponownie sprawdzić i potwierdzić uwzględnienie reguł biznesowych.

Formułowanie definicji celu i założeń wstępnych (1)

25

- Pierwszą fazą w procesie projektowania baz danych jest określenie definicji celu i założeń wstępnych.
  - Definicja celu ustala przeznaczenie bazy danych oraz określa konkretny punkt widzenia, który należy wziąć pod uwagę w trakcie prac projektowych.
- Każda baza danych stworzona jest w konkretnym celu np:
  - do rozwiązania konkretnego problemu biznesowego,
  - do obsługi transakcji konkretnej firmy lub organizacji,
  - jako część systemu informacyjnego.
- Definicja celu pozwala na identyfikację i określenie jej przeznaczenia.
  - W ten sposób łatwiej jest stworzyć odpowiednią strukturę bazy oraz zebrać dane niezbędne do wspierania jej założonego celu.

Formułowanie definicji celu i założeń wstępnych (2)

26

- Na tym etapie należy również sformułować założenia wstępne.
  - Te założenia reprezentują zadania, które w bazie danych będzie mógł wykonać użytkownik.
  - Założenia te powinny uzupełniać definicję celu i pomagać w określaniu poszczególnych elementów struktury bazy danych.
- Dwie odrębne grupy ludzi będą zaangażowane w tworzenie definicji celu oraz założeń wstępnych.
  - Pierwsza z nich, w której skład wchodzi projektant bazy, właściciel lub dyrektor organizacji oraz kierownictwo, odpowiedzialna jest za określenie definicji celu.
  - Druga, złożona z projektanta, kierownictwa oraz użytkowników bazy, zajmie się stworzeniem założeń wstępnych.

Analiza istniejącej bazy danych (1)

27

- Drugą fazą procesu projektowania bazy danych jest analiza obecnej bazy, jeśli taka istnieje.
  - W zależności od organizacji może to być baza spadkowa lub tradycyjna baza papierowa.
    - Baza spadkowa (zwana także odziedziczoną) to stara baza, użytkowana od wielu lat.
    - Papierowa baza danych to z kolei luźna kolekcja formularzy, kart indeksowych, teczek i tym podobnych.
- Bez względu na rodzaj lub stan bazy jej analiza dostarczy cennych informacji na temat sposobu wykorzystywania danych i zarządzania nimi przez organizację.
- Analiza pozwoli również na ocenę sposobu, w jaki organizacja gromadzi i prezentuje dane.
  - Należy przyjrzeć się sposobowi wykorzystania papieru do gromadzenia danych (w postaci formularzy), a także ich prezentacji (za pomocą raportów).
  - Jeśli do zarządzania i manipulacji danymi w bazie organizacja wykorzystuje jakieś oprogramowanie komputerowe, należy ocenić sposób gromadzenia przez nie danych i prezentowania ich na ekranie.
  - Należy również wziąć pod uwagę jak, jeśli w ogóle, organizacja korzysta z danych w sieci, oraz przyjrzeć się roli aplikacji internetowych współpracujących z bazą danych.

Analiza istniejącej bazy danych (2)

28

- Kolejnym elementem analizy jest przeprowadzenie rozmów z użytkownikami i kierownictwem w celu określenia sposobu, w jaki wykorzystują oni bazę danych na co dzień.
- Jako projektant musisz zapytać użytkowników o sposób wykorzystywania przez nich bazy oraz ich aktualne wymagania dotyczące informacji.
- Następnie przeprowadzasz rozmowę z personelem kierowniczym, pytając o informacje, które aktualnie otrzymuje, oraz o postrzeganie potrzeb informacyjnych organizacji.
- Te rozmowy są ważnym czynnikiem analizy, ponieważ pytania, które zadajesz (lub nie), będą miały duży wpływ na ostateczną strukturę bazy danych.
- Jeśli chcesz, by zaprojektowana przez Ciebie baza danych naprawdę spełniała potrzeby informacyjne organizacji, rozmowy, które przeprowadzasz, muszą być kompletne.

Analiza istniejącej bazy danych (3)

29

- Kolejnym krokiem jest wykorzystanie zebranych w trakcie analizy i rozmów informacji do stworzenia wstępnej listy pól.
- Następnie należy wprowadzić poprawki, usuwając wszystkie pola obliczeniowe i umieszczając je na osobnej liście — wykorzystane zostaną na późniejszym etapie procesu projektowego.
- Poprawiona lista zawiera podstawowe wymagania organizacji dotyczące danych, a także stanowi punkt wyjścia do stworzenia projektu nowej bazy danych.
- Z pewnością zawartość listy będzie stopniowo powiększana i modyfikowana w miarę rozwoju nowego projektu.
- Po sporządzeniu wstępnej listy pól należy wysłać ją użytkownikom i kierownictwu do wglądu i ewentualnej korekty.
  - Wszelkie komentarze powinny być przyjmowane z uwagą, a modyfikacje rozpatrywane.

Tworzenie struktur danych (1)

30

- Tworzenie struktur danych w bazie jest trzecim etapem procesu projektowego.
- Należy zdefiniować tabele i pola, określić klucze oraz specyfikację każdego z pól.
- Tabele są pierwszą strukturą bazy danych, którą należy zdefiniować.
  - Tematy, które będą reprezentowane w tabelach, określa się za pomocą listy założeń wstępnych, spisanych w trakcie pierwszej fazy procesu projektowego oraz wymogów dotyczących danych zebranych w trakcie fazy drugiej.
  - Następnie dla każdego z tematów należy utworzyć tabele oraz przypisać im odpowiednie pola z listy stworzonej w trakcie drugiej fazy projektowej.
  - Po zakończeniu tej czynności należy upewnić się, że każda tabela reprezentuje tylko jeden temat i nie zawiera duplikatów pól.

Tworzenie struktur danych (2)

31

- Teraz trzeba przyjrzeć się każdemu polu z osobna.
  - Upewnij się, że każde wieloczęściowe i wielowartościowe pole w tabeli zawiera tylko jedną wartość oraz że wszystkie pola, które nie zawierają cech wyróżniających dany temat, są usunięte.
- Następną czynnością jest przegląd i oczyszczenie struktur tabel.
  - To wymaga ponownego sprawdzenia pól, by upewnić się, że nic nie zostało przez przypadek pominięte, oraz kontroli struktur tabel.
- Kolejnym etapem jest ustalenie dla każdej tabeli odpowiednich kluczy.
  - Należy upewnić się, że każda tabela ma odpowiednio zdefiniowany klucz główny, który w sposób unikatowy identyfikuje każdy rekord tabeli.

Tworzenie struktur danych (3)

32

- Ostatnim krokiem tej fazy jest określenie specyfikacji każdego z pól w bazie danych
- Na tym etapie musisz przeprowadzić z użytkownikami bazy i kadrą kierowniczą rozmowy, które pomogą Ci zidentyfikować istotne dla nich specyficzne cechy charakterystyczne pól, oraz omówić cechy, o których mogą nie wiedzieć.
- Po przeprowadzeniu rozmów musisz określić i udokumentować specyfikacje dla każdego z pól w bazie.
- Następnie wraz z użytkownikami bazy i kierownictwem po raz kolejny przeglądasz strukturę tabeli i specyfikacje pól, by wprowadzić ewentualne poprawki.
- Po ich wprowadzeniu struktury tabel są gotowe do następnej fazy procesu.

Określanie i ustalanie zależności w tabelach

33

- Czwarta faza procesu projektowania baz danych polega na ustaleniu zależności w tabelach.
- Znowu należy przeprowadzić rozmowy z użytkownikami bazy oraz kadrą kierowniczą, określić cechy relacji oraz ustalić integralność na poziomie zależności.
  - Współpraca z użytkownikami i kierownikami jest niezwykle rozważnym posunięciem, ponieważ mogą Ci oni pomóc w identyfikacji zależności pomiędzy danymi.
  - Z pewnością nie są Ci znane wszystkie aspekty funkcjonowania danych w organizacji, uzyskanie tych informacji z pierwszej ręki przyniesie więc wiele korzyści.
- Kiedy relacje zostaną już ustalone, należy za pomocą kluczy głównych lub tabel łączących określić logiczne połączenia pomiędzy tabelami w każdej zależności.
  - To, z czego będziesz korzystać, zależy od rodzaju zależności ustalanej pomiędzy tabelami.
  - Następnie powinieneś określić typ i stopień uczestnictwa tabel w poszczególnych relacjach.
  - W niektórych przypadkach cechy te będą wynikać z rodzaju danych przechowywanych w tabelach. Czasem jednak trzeba się będzie oprzeć na konkretnych regułach biznesowych.

Określanie reguł biznesowych (1)

34

- Określanie reguł biznesowych jest piątym etapem procesu projektowania baz danych.
- W jego trakcie należy przeprowadzić rozmowy, zidentyfikować ograniczenia, jakim mają podlegać dane, określić reguły biznesowe oraz wprowadzić tabele walidacyjne.
  - Sposób, w jaki organizacja przegląda i wykorzystuje dane, narzuca wiele ograniczeń i wymagań, które będą musiały być uwzględnione w trakcie projektowania bazy danych.
- Rozmowy z użytkownikami i kierownictwem pomogą określić wymagania, którym powinny podlegać dane, ich struktury oraz zależności między nimi.
  - Stworzony na tej podstawie dokument posłuży jako zestaw reguł biznesowych.

Określanie reguł biznesowych (2)

35

- Rozmowy przeprowadzone z użytkownikami pozwolą odkryć specyficzne ograniczenia różnych aspektów bazy danych.
  - Na przykład użytkownik bazy danych przetwarzającej zamówienia ma świadomość tego, że data wysyłki musi być późniejsza niż data złożenia zamówienia, oraz że zawsze należy podać numer telefonu odbiorcy, a także wskazać sposób realizacji zamówienia.
  - Rozmowy z kierownictwem z kolei dadzą możliwość odkrycia ogólnych niedociągnięć w funkcjonowaniu bazy danych.
    - Na przykład menedżer agencji artystycznej wie, że agent może reprezentować nie więcej niż 20 artystów, a informacje promocyjne o każdym z artystów muszą być aktualizowane raz w roku.
  - Następnie należy zdefiniować i zaimplementować tabele walidacji, jeśli tylko okaże się to niezbędne dla wsparcia reguł biznesowych.
    - Jeśli ustalisz, że niektóre pola mają określoną liczbę wartości, spowodowaną sposobem wykorzystywania tych informacji przez organizację, możesz wykorzystać tabele walidacji do zapewnienia zgodności i logiczności wartości przechowywanych w tych polach.

Definiowanie widoków

36

- Szóstą fazą procesu projektowania baz danych jest definiowanie widoków.
- Po raz kolejny należy przeprowadzić konsultacje, zidentyfikować różne sposoby pracy z danymi oraz zdefiniować widoki.
- Aby zidentyfikować rodzaje widoków do zbudowania w nowej bazie należy dowiedzieć się, w jaki sposób użytkownicy i kierownicy pracują z danymi.
  - Może się okazać, że wielu pracowników, aby prawidłowo wykonać swoją pracę, będzie potrzebowało bardziej szczegółowych informacji, podczas gdy innym do podejmowania strategicznych decyzji wystarczą tylko dane ogólne.
  - Każda grupa użytkowników będzie wymagała innego typu dostępu, a umożliwić to mogą właśnie widoki.
- Następnym krokiem jest zdefiniowanie za pomocą odpowiednich tabel i pól widoków zidentyfikowanych w trakcie rozmów z pracownikami, a także określenie kryteriów dla tych widoków, które są niezbędne, by uzyskać dostęp do konkretnych informacji.
  - Określenie kryteriów będzie niezbędne w przypadku widoku pokazującego listę wszystkich klientów zlokalizowanych w Teksasie lub przedstawiającego ogólną liczbę autoryzowanych sprzedawców w Waszyngtonie, posegregowanych na podstawie miast.

Kontrola integralności danych (1)

37

- Siódmą i ostatnią fazą procesu projektowania baz danych jest kontrola struktury bazy pod kątem integralności danych.
- Najpierw należy przyjrzeć się po kolei każdej tabeli i wszystkim polom, by upewnić się, że spełniają one kryteria poprawności.
- Trzeba także rozstrzygnąć wszelkie nieścisłości oraz problemy i po raz kolejny sprawdzić struktury.
  - Po wprowadzeniu niezbędnych poprawek należy sprawdzić integralność na poziomie tabel.
- Następnie trzeba dokonać przeglądu specyfikacji wszystkich pól w bazie i po dokonaniu poprawek skontrolować integralność na poziomie pól.
  - W ten sposób upewnisz się, że ustalona i wprowadzona wcześniej integralność została zachowana.

Kontrola integralności danych (2)

38

- Po trzecie, należy sprawdzić poprawność każdej zależności, potwierdzić jej typ oraz atrybuty uczestnictwa wszystkich tabel występujących w zależnościach.
  - Należy przeanalizować integralność na poziomie zależności, aby upewnić się, że współdzielone pola zawierają pasujące do siebie wartości oraz że nie występują problemy z wprowadzaniem, aktualizowaniem lub usuwaniem danych w żadnej z tabel wchodzących w skład zależności.
- Na koniec należy przejrzeć stworzoną wcześniej listę reguł biznesowych i potwierdzić zgodność ograniczeń nałożonych na bazę danych.
  - Gdyby od czasu ostatnich rozmów z pracownikami pojawiły się nowe ograniczenia, należy określić je jako nowe reguły biznesowe i dopisać do istniejącej listy.
- Po zakończeniu całego procesu projektowania bazy danych logiczna struktura bazy będzie gotowa do implementacji przy użyciu systemu SZRBD.
  - **Procesu tego nie można jednak nigdy uznać za całkowicie ukończony, ponieważ struktura bazy danych zmienia się wraz z rozwojem organizacji.**

39

Tworzenie struktur tabel

Przegląd

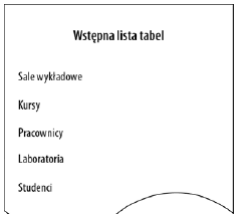
40

- Firmy korzystają z baz danych, by śledzić różne tematy, które są dla nich ważne.
  - Na przykład klinika medyczna prowadzi rejestr między innymi swoich pacjentów, lekarzy i terminów;
  - wypożyczalnia sprzętu musi utrzymywać dane na temat swoich klientów, sprzętu oraz umów najmu;
  - natomiast biuro dziekanatu jest zainteresowane (na najbardziej podstawowym poziomie) studentami, kadrą naukową i kursami.
- W każdym przypadku — i w każdym innym scenariuszu, który można sobie wyobrazić — tabela w bazie danych reprezentuje osobny podmiot.
- Ponadto każda tabela składa się z *pól* reprezentujących cechy, które definiują lub opisują podmiot tabeli.
- **Tabele stanowią fundament bazy danych**.
  - Fundament ten jest solidną i dobrą podstawą, gdy tabele są prawidłowo zaprojektowane.

Definiowanie wstępnej listy tabel

41

- Podczas tej części procesu projektowania bazy danych definiujemy *wstępną listę tabel*, którą wykorzystamy do zidentyfikowania i ustalenia tabel w nowej bazie danych.
- Do stworzenia tej listy używamy trzech procedur.
  - Pierwsza polega na wykorzystaniu *wstępnej listy pól*,
  - Druga polega na wykorzystaniu listy podmiotów zebranych podczas procesu przeprowadzania wywiadów,
  - Trzecia obejmuje wykorzystanie celów misji zdefiniowanych na początku procesu projektowania bazy danych.
- Kolejnym krokiem w tym procesie jest budowa struktury każdej tabeli z wykorzystaniem pól ze wstępnej listy pól.



Wstępna lista tabel

Sale wykładowe
Kursy
Pracownicy
Laboratoria
Studenci

Definiowanie ostatecznej listy

42

- Gdy wstępna lista tabel jest teraz tak kompletna, jak na to pozwala bieżący etap projektowania, można przystąpić do przekształcenia jej w *ostateczną listę tabel*.
- Nowa lista zawiera dwa elementy, których obecnie nie ma na wstępnej liście tabel: ***typ tabeli*** oraz ***opis tabeli***.
- Ostatnim zadaniem, jakie trzeba wykonać, aby przekształcić wstępną listę tabel w ostateczną listę tabel, jest dostrojenie nazw tabel.

Ostateczna lista tabel

Nazwa	Typ	Opis
Sale wykładowe	Dane	Przestrzenie lub obszary w obiekcie, zarezerwowane dla celów prowadzenia wykładów. Informacje dotyczące fizycznych aspektów, miejscowych zasobów i dostępności tych obszarów są przydatne, ponieważ pozwalają przypisać wykłady do obiektów, co może przyczynić się do bardziej optymalnego wykorzystania tych obszarów
Kursy	Dane	Programy nauczania prowadzone w tej instytucji w postaci oferowanych kursów. Informacje o kursach zawsze muszą odzwierciedlać dodawanie nowych kursów, usunięcie starych oraz ewolucję kursów istniejących

Typy tabel

43

- *Typ tabeli* pozwala na sklasyfikowanie tabeli według roli, jaką odgrywa ona w bazie danych, i zapewnia możliwości zidentyfikowania tabel działających w podobny sposób.
- Rola tabeli określa jej typ.
- Istnieją cztery typy tabel, które można przypisać do określonej tabeli.
  - *Tabela danych* reprezentuje podmiot, który jest istotny dla instytucji i jest podstawą informacji, które zapewnia baza danych.
  - *Tabela łącząca* ustanawia łącze pomiędzy dwoma tabelami w relacji wiele do wielu.
  - *Tabela typu podzbiór* zawiera pola, które są związane z określoną tabelą danych, i opisuje podmiot tabeli danych w bardzo dokładny sposób.
  - *Tabela walidacyjna* zawiera stosunkowo statyczne dane i jest kluczowym składnikiem integralności danych.

Opis tabeli

44

- *Opis tabeli* podaje czytelną definicję podmiotu reprezentowanego przez tabelę i określa, dlaczego podmiot ma istotne znaczenie dla instytucji.
- Istnieją pewne wytyczne, które regulują sposoby tworzenia opisów tabel.

Dostrajanie nazw tabel

45

- **Twórz unikatowe, opisowe nazwy, które mają sensowne znaczenie dla całej instytucji.**
  - Korzystanie z unikatowych nazw pomaga zapewnić, że każda tabela w czytelny sposób reprezentuje osobny podmiot i że wszyscy w instytucji rozumieją, co tabela przedstawia.
  - Należy wybierać nazwy, które są wystarczająco opisowe, aby nie było wątpliwości co do ich znaczenia. Remonty pojazdów to przykład dobrej, opisowej nazwy. Definiowanie unikatowej i opisowej nazwy wymaga trochę pracy, ale w dłuższej perspektywie ten wysiłek opłaci się.

Dostrajanie nazw tabel

46

- **Twórz nazwy, które dokładnie, czytelnie i jednoznacznie identyfikują podmiot tabeli.**
  - Niejasne lub wieloznaczne nazwy zazwyczaj wskazują, że tabela reprezentuje więcej niż jeden podmiot.
  - W przypadku napotkania takiej nazwy zidentyfikuj podmioty, które tabela naprawdę reprezentuje, i potraktuj każdy podmiot jako osobną tabelę. Dobrym przykładem niejasnej nazwy tabeli są Daty.
  - Nie ma możliwości, by dowiedzieć się, co tabela przedstawia, bez odwoływania się do jej opisu.
    - Załóżmy na przykład, że projektujemy bazę danych dla agencji muzycznej i nazwa *Daty* znalazła się na wstępnej liście tabel.
    - Po napotkaniu tej nazwy zdecydowaliśmy, że musimy zajrzeć do notatek sporządzonych w procesie wywiadu.
    - Odkryliśmy, że jedna z osób określiła nazwą Daty terminy spotkań z klientami, natomiast inna użyła tej samej nazwy do opisania dat zarezerwowanych dla artystów zrzeszonych w agencji.
    - Jest oczywiste, że ta tabela reprezentuje dwa podmioty. W związku z tym usuwamy nazwę Daty ze wstępnej listy tabel i zastępujemy ją dwoma nowymi tabelami: Spotkania z klientami oraz Harmonogramy artystów.

Dostrajanie nazw tabel

47

- **Używaj minimalnej liczby słów niezbędnych do przekazania znaczenia podmiotu tabeli.**
- Wszystkie osoby w instytucji powinny być w stanie określić, co przedstawia tabela, bez konieczności czytania jej opisu.
- Pomimo że naszym celem jest stworzenie krótkiej i zwięzłej nazwy tabeli, należy unikać minimalistycznego podejścia.
  - *TD_1* to dobry przykład nazwy, która jest zbyt krótka.
    - Nie będziemy mieć pojęcia, co reprezentuje ta tabela, o ile nie będziemy znali znaczenia wszystkich liter w nazwie.
- Należy również unikać podejścia skrajnie przeciwnego.
  - *Sprzęt samochodowy i remontowy różnego przeznaczenia* to nazwa zbyt długa i z łatwością można ją zastąpić nazwą *Sprzęt*.

Dostrajanie nazw tabel

48

- **Nie używaj słów, które przekazują cechy fizyczne.**
- Należy unikać używania w nazwach tabel słów takich jak plik, rejestr, karta czy tabela, ponieważ wprowadzają niepotrzebne zamieszanie.
  - Jest bardzo prawdopodobne, że tabela o nazwie zawierającej tego typu słowa reprezentuje więcej niż jeden podmiot.
- Weźmy za przykład nazwę Karta pacjenta. Na pozór może się wydawać, że taka nazwa jest prawidłowa. Wystarczy jednak pomyśleć trochę o tym, co reprezentuje karta pacjenta, aby zdać sobie sprawę, że jednak istnieją potencjalne problemy z tą nazwą.
  - Nazwa zawiera słowo, którego staramy się uniknąć (karta) i które potencjalnie reprezentuje trzy podmioty: pacjentów, lekarzy i wizyty. W związku z tym powinniśmy usunąć nazwę Karta pacjenta ze wstępnej listy tabel i zastąpić ją trzema nowymi nazwami — po jednej dla każdego z trzech podmiotów.

Dostrajanie nazw tabel

49

- **Nie należy używać akronimów i skrótów.**
- Akronimy są trudne do rozszyfrowania, skróty rzadko właściwie przekazują podmiot tabeli, a i jedne, i drugie łamią pierwszą wytyczną na tej liście. Rozważmy zastosowanie akronimów.
  - Załóżmy, że pomagamy jakiejś instytucji poprawić strukturę bazy danych i napotkaliśmy tabelę o nazwie SKK.
  - Skąd mamy wiedzieć, co reprezentuje tabela, nie wiedząc, co oznaczają poszczególne litery? Z pewnością nie można łatwo zidentyfikować podmiotu tabeli. Co więcej, może się okazać, że nazwa tabeli oznacza różne rzeczy dla różnych działów w instytucji. W związku z tym zdecydowaliśmy się przeprowadzić krótki wywiad z niektórymi pracownikami w celu stwierdzenia, co oznaczają litery. Okazało się, że pracownicy kadr uważają, że skrót SKK oznacza Studium Kształcenia Kadr, w dziale informatyki rozpoznają go jako System Kodów Kreskowych, natomiast w dziale bezpieczeństwa są pewni, że chodzi o Stanowisko Kierowania Kryzysem.
- Ten przykład wyraźnie pokazuje, dlaczego należy dołożyć wszelkich starań, aby unikać skrótów i akronimów w nazwach tabel.

Dostrajanie nazw tabel

50

- **Nie należy używać nazw własnych lub innych słów nadmiernie ograniczających dane, które mogą być wprowadzone do tabeli.**
- Wytyczna ta będzie nas chronić przed wpadnięciem w pułapkę tworzenia zdublowanych struktur tabel.
  - Na przykład nazwa Pracownicy z województwa pomorskiego poważnie ogranicza dane, które można wprowadzić do tej tabeli.
  - Co zrobimy z danymi pracowników z innych regionów, kiedy organizacja się rozwinie? Kiedy firma zacznie zatrudniać pracowników w Warszawie, Płocku czy Radomiu, będziemy zmuszeni stworzyć tabelę Pracownicy z województwa mazowieckiego, a kiedy pojawią się pracownicy z Koszalina, Kołobrzegu i Szczecina, trzeba będzie stworzyć tabelę Pracownicy z województwa zachodniopomorskiego.
- Zgodnie z zasadami projektowania baz danych nie należy tworzyć tego rodzaju duplikatów struktur, ponieważ mogą one stwarzać sporo problemów.

Dostrajanie nazw tabel

51

- **Należy używać formy nazwy w liczbie mnogiej.**
- Jak wiadomo, tabela reprezentuje jeden podmiot, który może być obiektem lub zdarzeniem.
- Idąc krok dalej, można stwierdzić, że tabela reprezentuje kolekcję złożoną z podobnych obiektów lub zdarzeń.
  - Na przykład przedstawiciel handlowy chce utrzymywać dane o wszystkich swoich klientach, a nie tylko jednym, a wypożyczalnia samochodów chce śledzić wszystkie swoje pojazdy, a nie tylko niebieskie bmw.
- Użycie formy liczby mnogiej nazwy jest dobrym pomysłem, ponieważ w czytelny sposób informuje, że zamierzamy odnosić się do kolekcji.
- Nazwy kolekcji, co oczywiste, zawsze występują w liczbie mnogiej (np. Łodzie, a nie Łódź).
- Natomiast słowa, które identyfikują pola, zawsze występują w liczbie pojedynczej (Numer telefonu, a nie Numery telefonów).
- Przestrzeganie tej zasady sprawi, że łatwo odróżnimy nazwy tabel od nazw pól w każdej dokumentacji stworzonej dla bazy danych (podczas nadawania nazw tabelom należy pamiętać, że w przypadku niektórych rzeczowników liczba pojedyncza i mnoga brzmią tak samo).

Wytyczne tworzenia nazw pól

52

- **Stosuj unikatowe, opisowe nazwy, które są zrozumiałe w całej instytucji.**
- Określona nazwa pola powinna występować tylko raz w całej bazie danych.
- Jedynym wyjątkiem od tej reguły jest sytuacja, gdy pole służy do ustanowienia relacji między dwiema tabelami.
- Należy się upewnić, że nazwa jest na tyle opisowa, aby właściwie przekazywała znaczenie wszystkim osobom, które ją zobaczą.
- Z ogólnymi nazwami pól, takimi jak Adres, Miasto i Województwo, można sobie poradzić poprzez użycie nazwy tabeli jako przedrostka nazwy pola.
  - W wyniku zastosowania tej techniki powstaną takie nazwy jak PracownikAdres, KlientAdres czy też DostawcaAdres.
  - W przypadku zastosowania takich nazw pól można skrócić prefiks (dla zapewnienia zwięzłości), używając pierwszych trzech lub czterech liter nazwy tabeli jako zmienionego prefiksu.
    - To pozwala przekształcić poprzednie nazwy pól na PracAdres, KltAdres, i DostAdres.

Wytyczne tworzenia nazw pól

53

- **Używaj nazw, które dokładnie, czytelnie i jednoznacznie identyfikują cechę, którą reprezentuje pole.**
  - Numer telefonu jest dobrym przykładem niedokładnej i niejednoznacznej nazwy pola. Jakiego rodzaju numer telefonu ma reprezentować pole? Domowy numer telefonu? Numer służbowy? A może numer telefonu komórkowego? Trzeba nauczyć się być konkretnym.
    - Jeśli mamy potrzebę zapisania każdego z tych rodzajów numerów telefonów, powinniśmy stworzyć trzy pola: Numer telefonu, Telefon do pracy i Telefon komórkowy.
- **Używaj minimalnej liczby słów niezbędnych do przekazania znaczenia cechy, którą reprezentuje pole**.
  - Należy unikać długich nazw pól, ale jednocześnie należy unikać stosowania nazw pól złożonych z jednego słowa, jeśli to słowo jest nieodpowiednie.
  - Na przykład: jeśli próbujemy zapisać datę rozpoczęcia pracy przez określonego pracownika, wówczas nazwa Zatrudniony jest zbyt krótka (i trochę niejasna), natomiast Data, kiedy pracownik został zatrudniony jest zbyt długa!
  - Data zatrudnienia jest jednak bardziej odpowiednią nazwą, która dokładnie przedstawia charakterystykę pola.

Wytyczne tworzenia nazw pól

54

- **Nie używaj akronimów, a ze skrótów korzystaj rozważnie.**
- Akronimy mogą być trudne do odczytania i często prowadzą do nieporozumień.
  - Wyobraźmy sobie pole o nazwie CAD_SW. Skąd mamy się dowiedzieć, co pole reprezentuje?
  - Z drugiej strony, możemy używać skrótów, o ile będziemy używać ich oszczędnie i obchodzić się z nimi ostrożnie.
  - Użycie skrótu jest uzasadnione tylko wtedy, gdy skrót uzupełnia lub poprawia nazwę pola w pozytywny sposób.
  - Zastosowanie skrótu nie powinno powodować, że nazwa staje się niejednoznaczna lub mniej czytelna.

Wytyczne tworzenia nazw pól

55

- **W nazwie używaj formy liczby pojedynczej.**
- Pole z nazwą w liczbie mnogiej, np. Umiejętności, sugeruje, że może zawierać dwie lub więcej wartości dla wskazanego rekordu, co nie jest dobrym pomysłem.
- Nazwa pola występuje w liczbie pojedynczej, ponieważ reprezentuje pojedynczą cechę podmiotu tabeli, do której należy.
  - Z kolei nazwy tabel występują w liczbie mnogiej, ponieważ reprezentują kolekcje podobnych do siebie obiektów lub zdarzeń.
  - Stosując tę konwencję nazewnictwa, można stosunkowo łatwo odróżnić nazwy tabel od nazw pól.

56

Klucze

- Istnieją cztery główne typy kluczy: *kandydujący*, *główny*, *obcy* oraz *nieistniejący*.
- Typ klucza określa jego funkcję w tabeli.

Dlaczego klucze są ważne

57

- Umożliwiają precyzyjną identyfikację każdego rekordu w tabeli.
  - Tabele reprezentują konkretne zbiory podobnych obiektów lub wydarzeń. Na przykład tabela Przedmioty reprezentuje zbiór przedmiotów, a nie tylko jeden przedmiot. Kompletny zestaw rekordów w tabeli składa się na zbiór, a każdy rekord reprezentuje pojedynczy element tego zbioru. Niezbędne są odpowiednie środki do jednoznacznej identyfikacji każdego elementu. Takim właśnie środkiem są klucze.
- Umożliwiają wprowadzenie i egzekwowanie różnych rodzajów integralności.
  - Klucze są ważnym składnikiem integralności na poziomie tabel oraz na poziomie relacji.
  - Dają na przykład gwarancję, że każda tabela będzie składać się z unikatowych rekordów, a pola wykorzystane do ustalenia zależności między dwiema tabelami zawsze będą miały identyczne wartości.
- Umożliwiają definiowanie zależności w tabelach.
  - Upewnij się, że klucze są zdefiniowane poprawnie dla każdej tabeli.
  - Umożliwi to zagwarantowanie zwartej struktury tabel, ograniczenie danych redundantnych oraz zapewni stabilność relacji między tabelami.

Klucze kandydujące

58

- Pierwszym rodzajem kluczy są klucze *kandydujące*.
- Jest to pole lub zestaw pól jednoznacznie identyfikujący pojedynczą instancję tematu tabeli.
- Każda tabela musi mieć *przynajmniej jeden* klucz kandydujący.
- Na późniejszym etapie przejrzymy dostępne w tabeli klucze kandydujące i wybierzemy jeden z nich jako klucz główny.
- Zanim konkretne pole będzie mogło zostać kluczem kandydującym, musi spełniać *wszystkie* poniższe warunki klucza kandydującego.
  - Warunki te składają się na wytyczne, którymi należy się kierować przy określaniu, czy dane pole będzie odpowiednie do pełnienia roli klucza kandydującego.
  - Pole nie może być kluczem kandydującym,jeżeli nie spełnia wszystkich poniższych warunków.

Warunki klucza kandydującego

59

1. Nie może być polem wieloczęściowym.
   - Widziałeś już, na czym polegają problemy związane z polami wieloczęściowymi, więc wiesz, że użycie takiego pola w charakterze identyfikatora byłoby złym pomysłem.
2. Musi zawierać unikatowe wartości.
   - Ta wartość zabezpiecza przed duplikowaniem zawartości rekordów w tabeli. Duplikowane rekordy są takim samym złem jak duplikowane pola i powinno się ich unikać za wszelką cenę.
3. Nie może zawierać znaczników null.
   - null oznacza brak wartości. Klucz kandydujący w żaden sposób nie może zidentyfikować rekordu, jeśli jego wartość wynosi „zero”.
4. Jego wartość nie może łamać zasad bezpieczeństwa lub prywatności narzuconych przez organizację.
   - Wartości takie jak hasła, numery polisy ubezpieczeniowej nie są odpowiednie do pełnienia funkcji klucza kandydującego.
5. Jego wartość nie może być w żadnej swojej części opcjonalna.
   - Opcjonalna wartość sugeruje, że dozwolona może być wartość zerowa. Można z tego wywnioskować, że wartość opcjonalna automatycznie stoi w opozycji do poprzedniego warunku i jako taka jest nieakceptowalna. To zastrzeżenie odnosi się głównie do sytuacji, w której funkcję klucza kandydującego pełnią dwa pola lub więcej.

Warunki klucza kandydującego (2)

60

6. Składa się z minimalnej liczby pól niezbędnej do uzyskania niepowtarzalności.
   - Funkcję klucza kandydującego może pełnić kombinacja pól (traktowana jako jedna wartość), pod warunkiem że każde z nich przyczynia się do utrwalenia niepowtarzalnej wartości całego klucza. Staraj się korzystać z jak najmniejszej liczby pól, ponieważ złożone klucze kandydujące mogą być trudne do ogarnięcia.
7. Jego wartości muszą jednoznacznie identyfikować każdy rekord w tabeli.
   - Ten warunek umożliwia ochronę przed duplikatami rekordów oraz zapewnia odwołanie do interesującego nas rekordu z innej tabeli w bazie.
8. Jego wartość musi bezpośrednio określać wartość każdego pola występującego w danym rekordzie.
   - Dzięki temu warunkowi klucze kandydujące tabeli są jedynym mechanizmem identyfikującym każdą wartość pola w danym rekordzie..
9. Jego wartość może być modyfikowana tylko w wyjątkowych przypadkach.
   - Nie powinieneś nigdy zmieniać wartości klucza kandydującego, jeżeli nie masz ku temu wyraźnych powodów. Nieprzemyślane zmiany mogą skutkować zaburzeniami zgodności z poprzednimi warunkami.

Sztuczne klucze kandydujące

61

- Jeśli tabela nie zawiera żadnych potencjalnych kluczy kandydujących, możesz stworzyć i wykorzystać sztuczny klucz kandydujący.
- Jest on sztuczny, ponieważ nie pojawia się w tabeli w sposób „naturalny”, należy go dopiero stworzyć.
- Można to zrobić, dodając do tabeli nowe pole, które spełnia wszystkie warunki klucza kandydującego.
- Takie pole staje się oficjalnym kluczem kandydującym tabeli.

Klucze główne

62

- **Klucz główny** reprezentuje tabelę w strukturze bazy danych i odgrywa ważną rolę w tworzeniu zależności między tabelami.
- **Wartość klucza głównego** w sposób jednoznaczny identyfikuje każdy rekord w tabeli oraz reprezentuje te rekordy w strukturze bazy danych. Chroni także przed powstawaniem duplikowanych rekordów.
- Klucz główny musi spełniać dokładnie te same wymagania co klucz kandydujący.
  - Jest on prosty do znalezienia, ponieważ wybiera się go z dostępnej puli kluczy kandydujących.
- Jeśli masz wybór między prostym (jednopolowym) kluczem kandydującym a kluczem złożonym, wybierz prosty.
  - Najlepiej jest korzystać z klucza, który zawiera możliwie jak najmniejszą liczbę pól.
- Wybierz klucz kandydujący, którego nazwa zawiera część nazwy tabeli.
  - Na przykład pole Nr faktury będzie dobrym kluczem kandydującym dla tabeli Faktury.

Zasady definiowania klucza głównego

63

- Każda tabela musi zawierać jeden — i tylko jeden — klucz główny.
  - Jeden klucz główny jest wystarczający, ponieważ musi on spełniać wszystkie niezbędne warunki.
- Wszystkie klucze główne w bazie danych muszą być unikatowe — żadne dwie tabele nie mogą zawierać identycznego klucza głównego, chyba że jedna z nich jest podzbiorem.
  - W związku z tym każda tabela musi posiadać własny klucz główny, by nie doszło do pomyłek lub niedomówień.
  - Tabele-podzbiory są zwolnione z tego obowiązku, ponieważ reprezentują konkretne wersje tematów tabel — w takim wypadku zarówno tabela główna, jak i tabela-podzbiór muszą dzielić ten sam klucz główny.

Klucze zastępcze

64

- Kiedy już wybierzesz odpowiedni klucz główny, pozostałe klucze kandydujące będą pełniły rolę kluczy *zastępczych*.
- Klucze te mogą być wykorzystane przez system SZRBD do jednoznacznego identyfikowania każdego rekordu w tabeli.
- Jeśli planujesz dalsze korzystanie z kluczy zastępczych, umieść obok każdego z nich oznaczenie KZ lub ZKZ (złożony klucz zastępczy).
  - W przeciwnym razie staną się one na powrót normalnymi polami.
  - Problem kluczy zastępczych nie będzie istotny w dalszych fazach procesu projektowego, znajdą one zastosowanie dopiero na etapie implementacji bazy danych w systemie SZRBD.

Klucze obce

65

- Klucz główny staje się kluczem obcym, kiedy użyjesz go do zdefiniowania między parą tabel relacji jeden-do-jednego albo jeden-do-wielu.
- Tak samo jak w przypadku każdego innego klucza, klucz obcy musi spełniaćokreślony zestaw reguł.
  - *Ma taką samą nazwę jak klucz główny, z którego został skopiowany.* Powinieneś postępować zgodnie z tą regułą, chyba że istnieje wyjątkowo ważny powód, aby tego nie robić.
  - *Korzysta z repliki specyfikacji pola klucza głównego, z którego został skopiowany.*

66

Specyfikacja pól

Dlaczego specyfikacje pól są ważne

67

1. Specyfikacje pól pomagają wprowadzić i utrzymać integralność na poziomie pól.
   - Wprowadzanie tych specyfikacji pozwala zagwarantować spójność i poprawność danych w polach.
2. Definiowanie specyfikacji pól wzmacnia ogólną integralność danych.
   - Pamiętaj, że integracja na poziomie pól jest jednym z czterech komponentów integralności. Wzmacnia ona (do pewnego stopnia) integralność na poziomie tabel, wprowadzoną na wcześniejszym etapie procesu projektowego. Stanie się to bardziej widoczne, kiedy zaczniesz pracować z logicznymi elementami specyfikacji pól.
3. Definiowanie specyfikacji pól obliguje Cię do głębokiego zrozumienia natury i celu wykorzystania danych w bazie.
   - Umożliwi Ci to określenie, czy dane te są rzeczywiście niezbędne dla organizacji, a także pomoże odpowiednio je wykorzystać.
4. Specyfikacje pól składają się na „słownik danych" bazy.
   - Każda specyfikacja przechowuje dane cech poszczególnego pola w bazie. Kompletny zestaw specyfikacji ustalony przez Ciebie dla wszystkich pól w bazie tworzy słownik struktury bazy danych. Ten słownik danych jest wyjątkowo przydatny podczas implementowania bazy w systemie SZRBD — możesz wykorzystać go jako przewodnik w trakcie tworzenia pól i ustalania ich podstawowych właściwości. Te specyfikacje pomogą także określić rodzaj procedur wprowadzania i walidacjidanych w trakcie tworzenia aplikacji bazodanowej.

Integralność na poziomie pól

69

- O *integralności pola* można mówić po zdefiniowaniu kompletnego zestawu specyfikacji dla danego pola.
- Integralność na poziomie pola gwarantuje, że:
  - Tożsamość i cel każdego pola jest jasny, a wszystkie tabele, w których ono występuje, są odpowiednio zidentyfikowane.
  - Definicje pól są spójne w całej strukturze bazy danych.
  - Wartości pól są spójne i poprawne.
  - Rodzaje modyfikacji, porównań i operacji na wartościach pola są jasno określone.

Anatomia specyfikacji pól

70

- Specyfikacje pól zawierają w sobie różnorodne elementy definiujące każdy atrybut pola.
- Wszystkie elementy specyfikacji dzielą się na następujące kategorie:
  - Elementy ogólne: Nazwa pola, Tabela-rodzic, Etykieta, Typ specyfikacji, Źródło specyfikacji, Dzielone z, Alias(y), Opis.
  - elementy fizyczne: Typ danych, Długość, Miejsca dziesiętne, Dozwolone znaki, Maska wprowadzania, Format wyświetlania
  - elementy logiczne: Typ klucza, Struktura klucza, Jednoznaczność, Wspieranie wartości zerowych, Wartości wprowadzone przez, Wartość wymagana, Wartość domyślna, Zakres wartości, Reguła wprowadzania, Dozwolone porównania, Dozwolone operacje.
- Te kategorie pozwalają skupić się na konkretnym aspekcie definiowanego pola, a także pomagają w odnajdywaniu elementów.

Przykład formularza służącego do definiowania specyfikacji pól

71

Ten element znajduje się jedynie w specyfikacji replikowanej i wskazuje nazwę konkretnej specyfikacji pola, na której oparta jest istniejąca specyfikacja.

Elementy ogólne

Nazwa pola:	Nr pracownika	**Typ specyfikacji:**	[x] Unikatowa [ ] Ogólna [ ] Kopia
Tabela-rodzic:	Pracownicy	**Specyfikacje źródłowe:**	
Etykieta:	Pracownik#		
Dzielony przez:	Pracownicy pełen etat, Pracownicy pół etatu, Klienci		
Zamiennik/zamienniki:			
Opis:	Unikatowy numer wykorzystywany do identyfikacji każdego pracownika w organizacji. Przydzielany jest w trakcie pierwszego dnia i nie zmienia się do końca trwania zatrudnienia		

Opis stanowi kompletną interpretację pola.

Jest to *najmniejsza* liczba słów unikatowo identyfikująca konkretne pole w bazie.

Tabela, która zawiera w swojej strukturze określone pole, nazywana jest *tabelą-rodzicem* tego pola. Jest to jedyna tabela, w której pole może się pojawić, chyba że jest ono częścią składową zależności.

Jest to alternatywna nazwa pola (zwykle krótsza), która może je identyfikować w aplikacji stworzonej dla bazy danych.

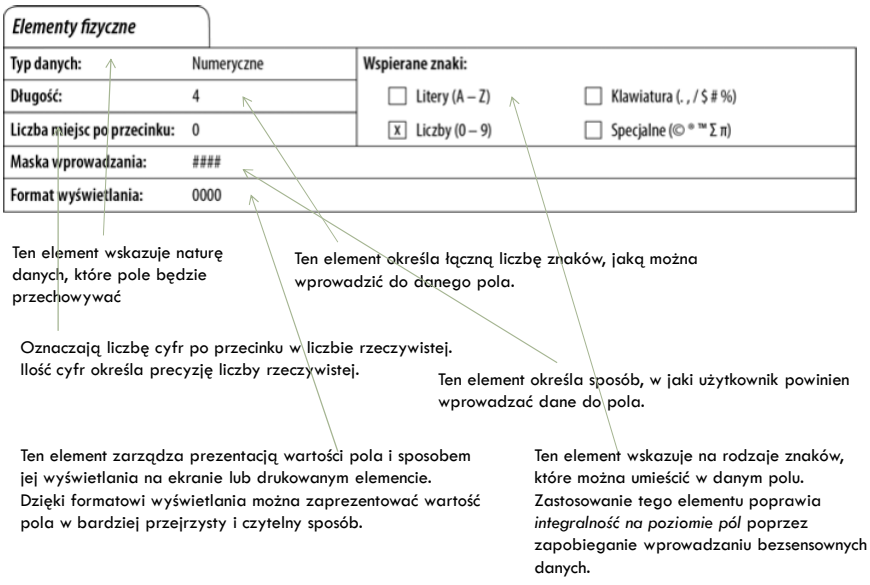
Ten element wskazuje nazwy innych tabel, które współdzielą to pole. Jedyne nazwy tabel, które powinny się tu pojawić, to te, które są w bezpośredniej zależności z tabelą-rodzicem danego pola.

Typ specyfikacji

72

- Unikatowa.
  - Jest to specyfikacja domyślna dla wszystkich pól poza tymi, które pełnią funkcję szablonów dla innych pól biorących udział w zależności jako klucze obce. Do tego typu specyfikacji należą wszystkie elementy oprócz specyfikacji źródłowej, a ich ustawienia będą miały zastosowanie jedynie do pola wskazanego przez element Nazwa pola.
- Ogólna.
  - Ta specyfikacja pełni rolę szablonu dla innych specyfikacji pól i pomaga stworzyć spójne definicje tych pól, których znaczenie ogólne jest takie samo. Możesz na przykład stworzyć ten typ specyfikacji dla ogólnego pola Miasto, a następnie użyć go jako podstawy dla innych pól w bazie oznaczonych jako Miasto. Pola takie jak Miasto klienta, Miasto pracownika, Miasto producenta mają wszystkie to samo znaczenie (reprezentują określone miasta), ale są wystarczająco różne, by można je było traktować jako osobne pola.
  - Specyfikacja ogólna wymaga wykorzystania niespecyficznych nazw pól i jak najbardziej ogólnych ustawień.
  - Nie mogą się w niej jednak znaleźć następujące elementy: Tabela-rodzic, Etykieta, Dzielone z, Alias(y) oraz Specyfikacja źródłowa.
- Replikowana.
  - Jest to specyfikacja domyślna dla pól bazujących na polach ogólnych lub takich, które w zależnościach pełnią rolę kluczy obcych, a większość ustawień czerpią z istniejących specyfikacji.
  - Możesz w niej zawrzeć elementy, których nie ma w specyfikacji źródłowej, a także zmodyfikować ustawienia pobrane z tej specyfikacji.

73



Typy danych (1)

74

- SQL definiuje osiem głównych typów danych, a każdy z nich ma jedną lub więcej unikatowych wariacji. Oto krótki opis każdego typu danych.
- Ciągi znaków.
  - Ten typ danych przechowuje ciągi znaków o stałej lub zmiennej długości, zawierające jeden lub wiele znaków nadających się do druku.
  - Ciąg znaków o stałej długości nazywany jest CHARACTER lub CHAR, a ten o zmiennej długości CHARACTER VARYING, CHAR VARYING lub VARCHAR.
- Ciągi znaków Unicode.
  - Ten typ danych jest podobny do poprzedniego, z tą różnicą, że przechowuje również znaki obcojęzyczne. Ciąg znaków o stałej długości nazywany jest NATIONAL CHARACTER, NATIONAL CHAR lub NCHAR, a ten o zmiennej długości NATIONAL CHARACTER VARYING, NATIONAL CHAR VARYING lub NCHAR VARYING.

Typy danych (2)

75

- Ciągi binarne.
  - Ten typ danych przechowuje dane binarne, takie jak zdjęcia, dźwięki, wideo, lub złożone dokumenty osadzone, takie jak pliki tekstowe lub arkusze kalkulacyjne.
  - Określa się je zwykle mianem BIT lub BIT VARYING.
- Dokładne dane liczbowe.
  - Ten typ danych przechowuje liczby całkowite i liczby dziesiętne.
  - Większość systemów SZRBD wdraża dokładne dane liczbowe jako NUMERIC, DECIMAL (DEC), INTEGER (INT), BIGINT lub SMALLINT, a każda wariacja określa zakres wartości akceptowanych przez pole.
- Przybliżone dane liczbowe.
  - Ten typ danych przechowuje liczby dziesiętne i wykładnicze.
  - Większość systemów SZRBD wdraża przybliżone dane liczbowe jako FLOAT, REAL lub DOUBLE PRECISION, a każda wariacja określa zakres wartości akceptowanych przez pole.

Typy danych (3)

76

- Logiczne dane liczbowe.
  - Ten typ danych przechowuje wartości, które można określić jako prawda lub fałsz, zwykle w formacie binarnym.
  - Niektóre systemy SZRBD do przechowywania tych danych wykorzystują BIT, INT lub TINYINT.
- Data i godzina.
  - Ten typ danych znany jest powszechnie większości systemów SZRBD jako DATE, TIME lub TIMESTAMP i przechowuje dane dotyczące dat, czasu i kombinacji tych dwóch wartości.
  - Pamiętaj, że implementacja tego typu danych różni się w zależności od wykorzystywanego systemu SZRBD. Aby wdrożyć je prawidłowo, musisz odwołać się do dokumentacji.
- Interwał.
  - Ten typ danych pokazuje ilość czasu zawartą pomiędzy dwiema wartościami typu danych DATE, TIME.
  - Wyrażony jest jako rok, miesiąc, rok/miesiąc dzień, godzina lub dzień/godzina.
  - Nie wszystkie systemy bazodanowe wspierają ten typ danych. Aby wdrożyć je prawidłowo, musisz odwołać się do dokumentacji.

77

- Wiele systemów SZRBD wprowadza dodatkowe typy danych, wykraczające poza standardy.
  - Nazywamy je rozszerzonymi typami danych.
  - Można do nich zaliczyć MONEY/CURRENCY (opisujące rodzaj waluty) oraz SERIAL/ROWID (identyfikujące rzędy).

Przykład formularza służącego do definiowania specyfikacji pól

78

Elementy logiczne

Typ klucza:	☐ Brak	☒ Główny	**Reguły edycji:**								
	☐ Obcy	☐ Alternatywny	☐ Wprowadź teraz, edycja dozwolona								
Struktura klucza:	☒ Prosta	☐ Złożona	☒ Wprowadź teraz, edycja niedozwolona								
Jednoznaczność:	☐ Niejednoznaczny	☒ Jednoznaczny	☐ Wprowadź później, edycja dozwolona								
Wspieranie wartości null:	☐ Null dozwolone	☒ Null niedozwolone	☐ Wprowadź później, edycja niedozwolona								
Wartości wprowadzane przez:	☐ Użytkownika	☒ System	☐ Nie można określić								
Wartość wymagana:	☐ Nie	☒ Tak									
Wartość domyślna:											
Zakres wartości:	1000–9999										
Dozwolone porównania:											
☒ To samo pole	☐ Wszystkie	☒ =	☐ >	☐ >=	☐ ≠	☐ <	☐ <=				
☐ Inne pola	☐ Wszystkie	☐ =	☐ >	☐ >=	☐ ≠	☐ <	☐ <=				
☒ Wyrażenie wartości	☐ Wszystkie	☒ =	☐ >	☐ >=	☐ ≠	☐ <	☐ <=				
Dozwolone operacje:											
☐ To samo pole	☐ Wszystkie	☐ +	☐ –	☐ x	☐ Π	☐ Łączenie					
☐ Inne pola	☐ Wszystkie	☐ +	☐ –	☐ x	☐ Π	☐ Łączenie					
☐ Wyrażenie wartości	☐ Wszystkie	☐ +	☐ –	☐ x	☐ Π	☐ Łączenie					

Pojęcia związane z wartością: Null

79

- Znacznik **null** wskazuje na brakującą lub nieistniejącą wartość.
- **null** to nie to samo co zero czy ciąg znaków zawierający jedno lub więcej pustych miejsc. Powody tego są bardzo proste:
  - Zero może mieć wiele różnych znaczeń. Może pokazywać stan konta, liczbę dostępnych biletów pierwszej klasy lub też ilość danego towaru w magazynie.
  - Mimo iż ciąg znaków zawierający jedno lub więcej pustych miejsc dla większości z nas z pewnością będzie niezrozumiały, ma on znaczenie w przypadku języka zapytań takiego jak SQL.
    - Puste miejsce jest w przypadku SQL poprawnym znakiem, a ciąg znaków złożony z trzech pustych miejsc (' ') jest równie prawidłowy jak ciąg znaków złożony z trzech liter ('abc').
  - Ciąg znaków o długości zerowej — dwa następujące po sobie pojedyncze cudzysłowy nie są przedzielone przerwą ('') — jest również akceptowalną wartością w językach takich jak SQL i może nabierać znaczenia w szczególnych okolicznościach.
    - Na przykład w tabeli PRACOWNICY wartość ciągu o długości zerowej w polu INICJAŁ DRUGIEGO IMIENIA może świadczyć o tym, że dany pracownik nie posiada drugiego imienia

Pojęcia związane z wartością: Null

80

Puste miejsce pokazuje, że do Bielska-Białej, Cieszyna oraz Brennej nie przypisano konkretnego kodu pocztowego

Klienci

Nr klienta	Imię klienta	Nazwisko klienta	Miasto	Kod pocztowy	<<inne pola>>
9001	Stanisław	Wojciechowski	Cieszyn		.....
9002	Zuzanna	Bartnicka	Kozy	43-340	.....
9003	Elwira	Rosińska	Dębowiec	43-426	.....
9004	Tomasz	Ekler	Wisła	43-460	.....
9005	Mateusz	Rawski	Bielsko-Biała		.....
9006	Katarzyna	Brzózka	Brenna		.....

Nieznana wartość może pojawić się w tabeli z wielu różnych powodów:
- specyficzna wartość, której potrzebujesz, dla danego pola pozostaje nieokreślona.
- żadna z wartości pola nie odnosi się do danego rekordu.

Problem ze znacznikami null

81

- Największą wadą znaczników null jest to, że mają one niekorzystny wpływ na działania matematyczne.
- Działanie zawierające null równa się null.
  - Jest to logiczne — jeśli dana liczba jest nieznana, wtedy wynik działania również pozostaje nieznany.
  - W poniższych przykładach widać, jak wartość null zmienia wynik działania:

(25 × 3) + 4 = 79

(Null × 3) + 4 = Null

(25 × Null) + 4 = Null

(25 × 3) + Null = Null

Problem ze znacznikami null

82

Wartość pola Razem wynika z wyrażenia matematycznego [Cena] x [Dostępna ilość]

Produkty

Nr produktu	Opis produktu	Kategoria	Cena	Dostępna ilość	Razem
70001	Zabezpieczenie rowerowe Shur-Lok	Akcesoria	75,00		
70002	Komputer rowerowy SpeedRite		65,00	20	1300,00
70003	Kask SteelHead Microshell	Akcesoria	36,00	33	1118,00
70004	Hamulce SureStop 133-MB	Komponenty	23,50	16	376,00
70005	Rower górski Diablo ATM	Rowery	1200,00		
70006	Lusterka na kask UltraVision		7,45	10	74,50

Kiedy wartość pola *Dostępna ilość* wynosi null, brakuje wartości pola Razem, czego rezultatem jest null występujący w polu Razem.

Prowadzi to do serii niewykrytych błędów, które pojawiają się, kiedy dodamy wszystkie wartości w kolumnie Razem, co da niedokładny wynik. **Błąd ten pozostaje niewykryty, ponieważ system SZRBD nie poinformuje Cię o nim**.

Problem ze znacznikami null

83

Podsumowanie kategorii

Kategoria	Ilość wystąpień
	0
Akcesoria	2
Rowery	1
Komponenty	1

- Rysunek przedstawia efekt, jaki znaczniki null mają na funkcje agregujące, które zawierają wartości danego pola w tabeli.
- Wynikiem funkcji agregującej, na przykład COUNT(<nazwapola>), będzie null, jeśli opiera się ona na polu również zawierającym wartość null.
- Rysunek pokazuje wyniki zapytania podsumowującego, które zlicza ogólną liczbę występowania poszczególnych kategorii z tabeli Produkty

Zapytanie podsumowujące pokazuje 0 wystąpień nieokreślonej kategorii, sugerując, że każdy produkt został przypisany do kategorii. Ta informacja jest niedokładna, ponieważ w tabeli Produkty znajdują się dwa produkty, które nie zostały przypisane do żadnej kategorii.

84

Związki między tabelami

Dlaczego związki są ważne

85

- Łączy ze sobą parę tabel, między którymi zachodzi logiczny związek.
  - Para tabel jest logicznie powiązana poprzez dane, jakie każda z nich zawiera.
- Pomaga lepiej zorganizować struktury tabel oraz ograniczyć redundancję danych.
  - Kiedy określasz relację między parą tabel, nieuchronnie wprowadzasz drobne zmiany w ich strukturach.
  - Takie udoskonalenia sprawiają, że struktury tabel stają się wydajniejsze, oraz minimalizują ilość redundantnych danych, które mogą znajdować się w tabelach.
- Stanowi mechanizm, który pozwala na pobieranie danych z wielu tabel jednocześnie.

Pojęcia związane z zależnościami: zależności

86

- Zależność istnieje pomiędzy dwiema tabelami, kiedy można w jakiś sposób powiązać rekordy z pierwszej tabeli z rekordami z drugiej.
  - Można ustalić zależność poprzez zestaw kluczy podstawowych i obcych lub wykorzystując trzecią tabelę zwaną łączącą (lub asocjacyjną).
  - Sposób, w jaki ustala się zależność, zależy od typu zależności, jaka istnieje pomiędzy tabelami.
- Zależność jest istotnym komponentem relacyjnej bazy danych.
  - Pozwala na stworzenie widoku wielotabelowego.
- Jest niezbędna do zachowania integralności danych, ponieważ pomaga zredukować zbędne dane i wyeliminować duplikaty.
- Każdą zależność można scharakteryzować na trzy sposoby: poprzez **typ zależności**, jaka istnieje pomiędzy tabelami, **sposób**, w jaki tabele biorą udział w zależności, oraz **stopień**, w jakim to robią.

Ustanawianie charakterystyk relacji

87

- Charakterystyki te określają, co się stanie, kiedy usuniesz rekord, rodzaj udziału każdej tabeli w relacji oraz stopień, w jakim każda z tabel uczestniczy w relacji.

Definiowanie reguły usuwania dla każdej relacji

88

- Reguła ta określa, co Twój SZRBD danych powinien zrobić, kiedy przedstawisz żądanie usunięcia danego rekordu z nadrzędnej tabeli w relacji.
- Reguły usuwania są kluczowe z punktu widzenia integralności na poziomie relacji, ponieważ pomagają zapobiegać powstawaniu *osieroconych* rekordów, czyli rekordów w tabeli podrzędnej, które nie mają żadnego powiązania z rekordami w tabeli nadrzędnej.
- Istnieje pięć typów reguł usuwania, które możesz zdefiniować, oraz działań, jakie powinien podjąć SZRBD, kiedy obowiązuje dana reguła.

89

- 1. Odmowy — SZRBD nie usunie rekordu w tabeli nadrzędnej, tylko go zachowa i oznaczy jako „nieaktywny”.
- 2. Restrykcyjna — SZRBD nie usunie rekordu w tabeli nadrzędnej, jeśli w tabeli podrzędnej istnieją powiązane z nią rekordy. SZRBD powinien najpierw usunąć wszystkie powiązane rekordy z tabeli podrzędnej, zanim będzie można usunąć rekord w tabeli nadrzędnej.
- 3. Kaskadowa — SZRBD podejmie dwie określone akcje: usunie rekord z tabeli nadrzędnej, a także automatycznie usunie wszystkie powiązane rekordy z tabeli podrzędnej.
- 4. Wartości null — SZRBD usunie rekord z tabeli nadrzędnej, po czym nada polom klucza obcego w powiązanych rekordach tabeli podrzędnej wartość null. Jeśli zamierzasz korzystać z tej reguły usuwania, powinieneś zmienić specyfikację pola klucza obcego i w elemencie Wspieranie wartości null zaznaczyć „Null dozwolone”.
- 5. Wartości domyślnej — SZRBD usunie rekord z tabeli nadrzędnej, a następnie nada polom klucza obcego w powiązanych rekordach tabeli podrzędnej bieżącą wartość domyślną podaną w parametrach elementu logicznego specyfikacji pola klucza obcego. Oczywiście aby korzystać z tej reguły, musisz zdefiniować w specyfikacji element Wartość domyślna.

90

Sprawdzanie i korygowanie integralności danych

Integralność na poziomie tabel i pól

91

- Aby uzyskać pewność osiągnięcia integralności na poziomie tabel, należy przejrzeć projekty wszystkich tabel i upewnić się, że spełniają one wszystkie wymagania z poniższych punktów:
  - W żadnej z tabel nie ma duplikatów pól.
  - W żadnej z tabel nie ma pól obliczeniowych.
  - W żadnej z tabel nie ma pól wielowartościowych.
  - W żadnej z tabel nie ma pól wieloczęściowych.
  - W żadnej z tabel nie ma duplikatów rekordów.
  - Każdy rekord w tabeli jest identyfikowany przez wartość klucza głównego.
  - Każdy klucz główny spełnia warunki klucza głównego.
- Aby upewnić się, że osiągnęliśmy integralność na poziomie pól, powinniśmy wykonać następujące czynności:
  - sprawdzić, czy wszystkie pola spełniają *warunki idealnego pola*;
  - sprawdzić, czy dla każdego pola zdefiniowano zbiór specyfikacji.

Integralność na poziomie referencji

92

- Aby upewnić się, że osiągnęliśmy integralność na poziomie związków, powinniśmy sprawdzić wszystkie związki pomiędzy tabelami.
- Ten poziom integralności został osiągnięty, jeśli zrealizowaliśmy następujące zadania:
  - właściwie utworzyliśmy relacje;
  - zdefiniowaliśmy właściwe reguły usuwania rekordów;
  - prawidłowo zidentyfikowaliśmy typ uczestnictwa dla każdej tabeli;
  - prawidłowo określiliśmy stopień uczestnictwa dla każdej tabeli.

Bibliografia

93

- Bazy danych - Podstawy projektowania i języka SQL. Krystyna Czapla, Helion 2015.
- Projektowanie baz danych dla każdego. Michael J.Hernandez, Helion 2014.
- Antywzorce języka SQL. Bill Karwin, Helion 2012.
- http://wazniak.mimuw.edu.pl/images/a/a7/BD-2st-1.2-w04.tresc-1.1.pdf