

» Idź do

- Spis treści
- Przykładowy rozdział

» Katalog książek

- Katalog online
- Zamów drukowany katalog

» Twój koszyk

- Dodaj do koszyka

» Cennik i informacje

- Zamów informacje o nowościach
- Zamów cennik

» Czytelnia

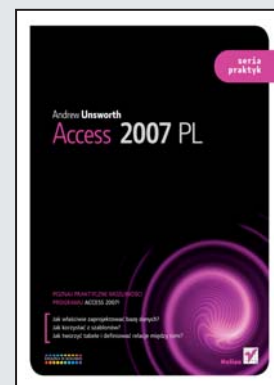
- Fragmenty książek online

» Kontakt

Helion SA
ul. Kościuszki 1c
44-100 Gliwice
tel. 032 230 98 63
e-mail: helion@helion.pl
© Helion 1991-2008

Access 2007 PL. Seria praktyk

Autor: Andrew Unsworth
Tłumaczenie: Radosław Meryk
ISBN: 978-83-246-2060-9
Tytuł oryginału: [Access 2007 in Easy Steps \(In Easy Steps\)](#)
Format: 180x235, stron: 200



Poznaj praktyczne możliwości programu Access 2007!

- Jak właściwie zaprojektować bazę danych?
- Jak korzystać z szablonów?
- Jak tworzyć tabele i definiować relacje między nimi?

Wbrew pozorom nie trzeba być specjalistą, żeby korzystać z Accessa! Jest to program wyjątkowo przyjazny dla użytkownika, umożliwiający tworzenie baz danych i zarządzanie nimi bez potrzeby dogłębnego poznawania języka SQL oraz skomplikowanych środowisk serwerowych. Aplikacja pozwala na zapisywanie danych z wykorzystaniem formularzy, kierowanie zapytań do bazy, a także dzielenie danych ze współpracownikami za pośrednictwem sieci komputerowej.

Książka „Access 2007 PL. Seria praktyk” zawiera zwięzły i czytelny opis wszystkich najważniejszych funkcji tego programu, a także konkretne przykłady i jasne instrukcje zastosowania narzędzi Accessa. Kolorowe strony pozwalają na szybkie odnalezienie interesujących Cię zagadnień. Dzięki temu podręcznikowi poznasz podstawowe zasady tworzenia dobrego projektu bazy danych oraz jej zaawansowane możliwości. Nauczysz się tworzyć tabele, formularze i raporty, a także korzystać z kluczy podstawowych i obcych. Bez problemu zbudujesz taką bazę danych, która pozwoli Ci sprawnie zarządzać informacjami.

- Personalizacja Accessa 2007
- Projektowanie baz danych
- Relacyjne bazy danych
- Klucze podstawowe i obce
- Tworzenie tabel
- Korzystanie z typów danych
- Definiowanie relacji
- Kwerendy
- Korzystanie z SQL
- Tworzenie i dostrajanie formularzy
- Tworzenie raportów
- Współdzielenie Accessa

Naucz się korzystać z Accessa – zachwyca Cię jego możliwości!

1

Zaczynamy

7

Czym jest Access 2007?	8
Ekran początkowy	9
Pasek narzędzi Szybki dostęp	10
Korzystanie z Przycisku pakietu Office	12
Personalizacja Accessa 2007	15
Konwersja starszych baz danych	16
Korzystanie z serwisu Office Online	18
Szkolenia online	19
Pobieranie nowych materiałów	20
Pobieranie nowych szablonów	21
Korzystanie z szablonów	22
Korzystanie ze wstążki	24
Korzystanie z Okienka nawigacji	25
Korzystanie z systemu pomocy	26

2

Projektowanie baz danych

27

Relacyjne bazy danych	28
Typy relacji	30
Projekt bazy danych	32
Zanim uruchomisz Accessa	33
Projektowanie tabel	34
Klucze podstawowe i obce	35
Doskonalenie projektu	36

3

Tworzenie tabel

37

Okno tabeli	38
Korzystanie z szablonów tabel	39
Korzystanie z widoku arkusza danych	40
Dodawanie i usuwanie pól	41
Typy danych w widoku arkusza danych	43
Korzystanie z widoku projektu	44
Tworzenie tabeli	45
Wstawianie wiersza w widoku projektu	46
Usuwanie pola w widoku projektu	47
Określanie klucza podstawowego	48
Korzystanie z typów danych	49
Korzystanie z załączników	51

Określanie właściwości pól	52
Korzystanie z reguł weryfikacji poprawności danych	53
Tworzenie masek wprowadzania	54
Ustawianie domyślnej wartości	55
Korzystanie z indeksów	56
Tworzenie kolumny odnośnika	58

4

Definiowanie relacji

61

Okno Relacje	62
Dodawanie tabel	63
Definiowanie relacji	64
Więzy integralności	66
Określanie właściwości sprzężenia	68
Sprzężenia lewo- i prawostronne	70

5

Praca z danymi

71

Wprowadzanie danych	72
Wstawianie wierszy	73
Korzystanie ze Schowka	74
Kopiowanie i wklejanie danych	75
Kopiowanie danych bezpośrednio z Excela	76
Kopiowanie danych do Excela	77
Importowanie danych z Excela	78
Importowanie danych z Accessa	83
Połączenia z bazami danych Accessa	85
Zarządzanie zadaniami importowania	87
Zbieranie danych za pośrednictwem poczty elektronicznej	88
Filtrowanie danych	94
Podsumowania kolumn	96
Sprawdzanie pisowni danych	97
Formatowanie danych	98

6

Kwerendy

101

Czym jest kwerenda?	102
Korzystanie z kreatora kwerend	103
Widok projektu kwerendy	106
Dodawanie kryteriów do kwerend	108
Kwerendy tworzone na podstawie wielu tabel	109
Definiowanie kryteriów dla liczb	110
Definiowanie kryteriów dla danych tekstowych	111
Definiowanie kwerend tworzących tabele	112
Definiowanie kwerend dołączających	114
Definiowanie kwerend aktualizujących	115
Definiowanie kwerend usuwających dane	116

7

Korzystanie z SQL

117

Posługiwanie się językiem SQL	118
Trzy odmiany języka SQL	119
Otwieranie okna SQL	120
Korzystanie z klauzuli SELECT	121
Korzystanie z klauzuli WHERE	122
Funkcje agregacji języka SQL	123
Tworzenie kwerend składających	124

8

Tworzenie formularzy

125

Czym jest formularz?	126
Anatomia formularza	127
Wykorzystanie formularzy do wprowadzania danych	128
Filtrowanie formularzy	130
Zastosowanie szczegółowego filtra	131
Korzystanie z kreatora formularzy	132
Tworzenie prostego formularza	135
Używanie dzielonych formularzy	136
Korzystanie z formularzy zawierających wiele elementów	139
Wyszukiwanie rekordów	140

9

Dostrajanie formularzy

141

Korzystanie z widoku projektu	142
Korzystanie z widoku układu	143
Korzystanie z narzędzia Lista pól	144
Dodawanie nagłówek i stopek	145
Dodawanie formantów do formularza	146
Dostrajanie formantów	147
Zmiana właściwości formantów	148
Tworzenie formantów wyliczanych	149
Zmiana kolejności dostępu	150
Tworzenie formularzy z zakładkami	151
Wykorzystanie przycisków	154
Tworzenie modalnych okien dialogowych	156
Używanie makr	160

10

Tworzenie raportów

161

Korzystanie z kreatora raportów	162
Tworzenie prostego raportu	166
Korzystanie z widoku projektu raportu	167
Dodawanie pól do raportu	168
Dodawanie formantów do raportu	169
Dodawanie nagłówek i stopek	170
Sortowanie i grupowanie danych	171
Drukowanie etykiet	172
Ustawianie niestandardowych rozmiarów etykiet	175
Korzystanie z podglądu wydruku	176
Drukowanie raportów	177
Wysyłanie raportów pocztą elektroniczną	178

11

Współdzielenie Accessa

179

Ochrona baz danych za pomocą haseł	180
Tworzenie baz danych ACCDE	182
Wykonywanie kopii zapasowych baz danych Accessa	183
Podział bazy danych	184
Interakcje z programem SharePoint	186

Skorowidz

187

2

Projektowanie baz danych

Prawidłowy projekt jest kluczem do sukcesu i przydatności bazy danych. Trochę czasu poświęconego na projekt w odpowiednim momencie pozwoli uniknąć poprawiania błędów na późniejszym etapie.

- 28 Relacyjne bazy danych**
- 30 Typy relacji**
- 32 Projekt bazy danych**
- 33 Zanim uruchomisz Accessa**
- 34 Projektowanie tabel**
- 35 Klucze podstawowe i obce**
- 36 Doskonalenie projektu**

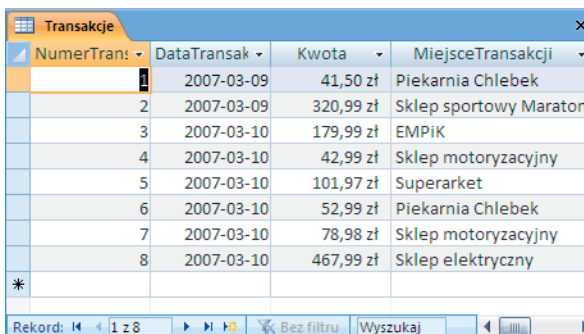
Relacyjne bazy danych

Jest zupełnie naturalne, że w zetknięciu z niemiłą perspektywą nauki nowego pakietu programowego większość osób głośno krzyczy i w ślepej panice szuka najbliższego wyjścia. Dotyczy to również nowicjuszy w świecie relacyjnych baz danych. Wcale jednak nie musi tak być. Mówiąc prosto, relacyjna baza danych jest kolekcją tabel zawierających informacje powiązane ze sobą za pomocą wspólnych pól w taki sposób, aby można było z nich korzystać bardziej efektywnie. Pojęcie „relacyjna” odnosi się do sposobu modelowania danych. To dzięki tej właściwości osiągnąć większą efektywność. Wiele nieporozumień wynika z różnej interpretacji pojęć. Powodem tego stanu w dużym stopniu jest liberalne stosowanie w branży żargonu. Większość terminów jest używana zamiennie.

W niniejszym rozdziale omówimy różne pojęcia występujące w modelu relacyjnym oraz przedstawimy plan działań przy projektowaniu baz danych, tak by podzielić pracę na mniejsze, łatwiejsze do przełknięcia „kąski”.

Co to jest relacja?

Pomimo technicznie brzmiącej nazwy relacja nie jest niczym innym, jak tabelą danych, podobną do zaprezentowanej na poniższym zrzucie ekranu.



NumerTran	DataTransak	Kwota	MiejsceTransakcji
1	2007-03-09	41,50 zł	Piekarnia Chlebek
2	2007-03-09	320,99 zł	Sklep sportowy Maraton
3	2007-03-10	179,99 zł	EMPIK
4	2007-03-10	42,99 zł	Sklep motoryzacyjny
5	2007-03-10	101,97 zł	Superarket
6	2007-03-10	52,99 zł	Piekarnia Chlebek
7	2007-03-10	78,98 zł	Sklep motoryzacyjny
8	2007-03-10	467,99 zł	Sklep elektryczny
*			

Aby bazy danych Accessa mogły być używane wydajniej, zawierają więcej niż jedną tabelę. Jak się przekonamy później, aby bazę danych można było uznać za relacyjną, musi ona zawierać co najmniej dwie tabele. W niniejszej książce zamiast pojęcia „relacja” będzie używane mniej sformalizowane pojęcie „tabela”.

W relacyjnej bazie danych tabela opiera się na określonym podmiocie (jednostce). Zazwyczaj dotyczy ona konkretnego obiektu, na przykład samochodu, i zawiera dane specyficzne dla tego obiektu. Na przykład, w bankowej bazie danych może być tabela *Klient* zawierająca dane klientów tego banku.

dokończenie...

Inną tabelą, jaka może znaleźć się w bankowej bazie danych, jest *Rachunek* — tabela zawierająca informacje o typie rachunku posiadanego przez określonego klienta wraz z bieżącym saldem.

Tabela zawiera **kolumny**, których bardziej formalna nazwa to **poła**, oraz **wiersze** — bardziej oficjalnie nazywane **rekordami**. Logiczne związki występujące pomiędzy tabelami są określane terminem **relacje**.

Poła

Pole to techniczna nazwa kolumny. Służy ono do opisanego rodzaju danych. Na przykład pole *Płeć*, którego definicję zaprezentowano poniżej, przedstawia płeć klienta. W tabeli *Samochód* mogłoby się znaleźć pole *Kolor* opisujące kolor samochodu. Chociaż niektórzy mogą skłaniać się do określania pola terminem kolumna, może to powodować mylenie pól z innymi obiektami. W niniejszej książce będziemy używali pojęcia **poła**.

Płeć ▾
Mężczyzna
Mężczyzna
Kobieta
Mężczyzna
Kobieta
Mężczyzna

Rekordy

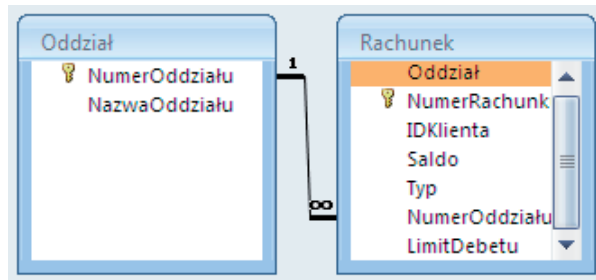
Rekord nieformalnie określany jest jako wiersz i zawiera właściwe dane zapisane w tabeli. O ile pole opisuje rodzaj danych w tabeli, na przykład płeć klienta, o tyle rekord informuje, czy wybrany klient jest mężczyzną, czy kobietą. Jeśli tabelę uznamy za opis obiektu, na przykład samochodu, to wiersze tabeli będą prezentowały egzemplarze poszczególnych samochodów.

Samochód			
NumerReje: ▾	Model ▾	Pojemność: ▾	Skrzynia bie ▾
GSZ L372	Terenowy	4,5	Automatyczna
ELE 2545	Terenowy	3,0	Automatyczna
WZW 1245	Rodziny	1,6	Manualna
GDY 3678	Sportowy	2,5	Manualna
KAT 3467	Mini	1,0	Automatyczna

Typy relacji

Bez relacji baza danych nie byłaby relacyjna. To wydaje się oczywiste. Czym jednak dokładnie są relacje? I dlaczego są one tak ważne?

Relacja jest logicznym połączeniem pomiędzy tabelami. Połączenie tworzy się pomiędzy polem w jednej tabeli a polem w innej tabeli. Dla przykładu, poniżej zaprezentowano dwie tabele: *Oddział* i *Rachunek*.



W jednym oddziale banku jest wiele rachunków klientów. Wiążąc pole *NumerOddziału* w tabeli *Oddział* z polem *NumerOddziału* w tabeli *Rachunek*, tworzy się pomiędzy tymi tabelami relację jeden do wielu.

Pomiędzy tabelami można zdefiniować trzy różne typy relacji: „jeden do jednego”, „jeden do wielu” oraz „wiele do wielu”. Każdą z nich pokrótce opisano poniżej.

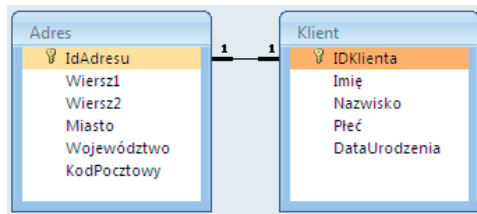
Jeden do wielu

O tym typie relacji wspomniano w poprzednim punkcie. Korzysta się z niej w przypadku, gdy rekordowi jednej z tabel, określanej terminem **rodzic** (w poprzednim przykładzie tę rolę spełniała tabela *Oddział*), odpowiada więcej niż jeden rekord w innej tabeli, znanej jako **dziecko** (w przykładzie — tabela *Rachunek*).

Jeden do jednego

Relacja tego typu występuje w przypadku, gdy istnieje bezpośrednie powiązanie pomiędzy rekordem w jednej tabeli a rekordem w innej. Przykładowo, na początku następnej strony zaprezentowano dwie tabele: *Klient* i *Adres*. Zgodnie z logiką reguł biznesu, powiązaną z tą relacją, wybrany klient może w danym momencie mieć tylko jeden adres. Z tego względu każdemu rekordowi w tabeli *Klient* odpowiada dokładnie jeden rekord w tabeli *Adres*.

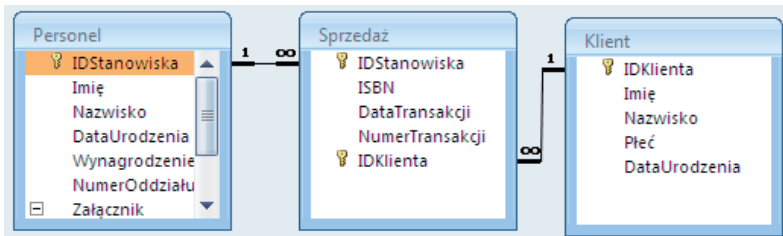
dokończenie...



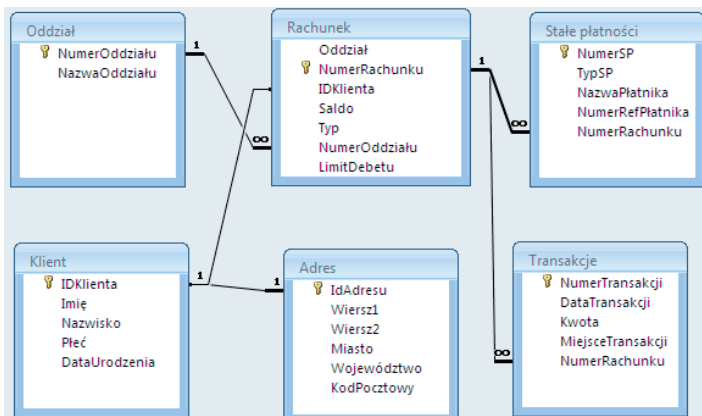
Pomiędzy powyższymi tabelami zachodzi relacja jeden do jednego.

Wiele do wielu

W relacji wiele do wielu rekord w jednej tabeli może mieć wiele odpowiedników w drugiej tabeli, i na odwrót. W celu zamodelowania tego typu relacji w Accessie należy utworzyć trzecią tabelę, zwaną **tabelą łączącą**. Pełni ona rolę pośrednika, który pozwala na zredukowanie relacji wiele do wielu do dwóch relacji jeden do wielu. Przykład zastosowania tabeli łączącej zaprezentowano poniżej.



Na poniższej ilustracji pokazano relacje występujące w typowej bazie danych.



Projekt bazy danych

Dobry projekt bazy danych ma kluczowe znaczenie dla utworzenia bazy danych pozwalającej na dokładne i wydajne przechowywanie i pobieranie informacji. Na szczęście, nauczenie się podstawowych zasad projektowania baz danych i stosowanie ich do własnych baz danych nie jest trudne.

W niniejszym podrozdziale omówimy zagadnienia, które są niezbędne do samodzielnego projektowania baz danych, gdyż kompletny opis wszystkich zasad projektowania baz danych z powodzeniem zajęłoby oddzielną książkę.

Korzyści wynikające z dobrego projektu baz danych

Istnieje wiele powodów, dla których warto poświęcić trochę czasu na uważne zaplanowanie i przygotowanie projektu nowej bazy danych. Pozwala to nie tylko skoncentrować się na problemie, który próbujemy rozwiązać, lecz także zmniejszyć liczbę błędów popełnianych podczas pracy z bazą danych. Tym samym oszczędza to wielu kłopotów. Naprawa źle zaprojektowanej bazy danych może zająć wiele cennego czasu. Z kolei dobry projekt baz danych może przyczynić się do:

- zmniejszenia ilości redundantnych danych;
- zmniejszenia rozmiaru plików bazy danych;
- zwiększenia wydajności bazy danych Accessa;
- zapewnienia dokładności danych zapisanych w bazie.

Dlaczego warto dążyć do zmniejszenia ilości redundantnych danych?

Choć niektórzy nie przywiązują do redundancji zbyt wielkiej wagi, należy za wszelką cenę dążyć do tego, by redundantnych danych było jak najmniej. Redundantne dane to informacje, które albo występują w innej tabeli, albo takie, które Access może obliczyć, zatem w ogóle nie ma potrzeby zapisywania ich w bazie.

Na przykład, w tabeli może występować pole przeznaczone na datę urodzenia klienta oraz pole na jego wiek. W tym przypadku pole dla wieku jest zbędne, ponieważ wiek klienta można obliczyć, odejmując datę urodzenia klienta od daty bieżącej.

Nieco większy rozmiar pliku nie wydaje się sprawą szczególnie szkodliwą. Istotnie, jeśli w bazie danych nie ma zbyt wielu informacji, nie jest to trudność. Jednak wraz ze zwiększeniem objętości danych zapisanych w bazie wzrasta ilość czasu, jakiego Access potrzebuje na wykonanie zapytania lub wygenerowanie raportu.

Zanim uruchomisz Accessa

Cykl życia projektu bazy danych

Każdy projekt wymaga planu. Tworzenie bazy danych Accessa nie jest pod tym względem wyjątkowe. Dlatego właśnie jeśli nikt nie straszy terminami, projektanci baz danych stosują się do cyklu życia projektu bazy danych. Cykl taki przebiega następująco:

- identyfikacja wymagań;
- projekt baz danych;
- utworzenie bazy danych w Accessie;
- utworzenie formularzy i raportów;
- testowanie.

W tym rozdziale przyjrzymy się pierwszym trzem etapom cyklu życia projektu, ponieważ te elementy mają największy wpływ na dokładność i wydajność bazy danych. Podczas pracy z niniejszą książką wyrobisz sobie własny pogląd na to, w jaki sposób powinny działać formularze oraz jakie kwerendy należy zdefiniować.

Identyfikacja wymagań

Pierwszą czynnością, jaką należy wykonać w ramach projektowania bazy danych, jest zidentyfikowanie jej wymagań. Sprowadza się to do zadania sobie następującego pytania: „Jakie czynności powinna wykonywać baza danych?”. Na pierwszy rzut oka pytanie może wydawać się oczywiste, jednak znane są przypadki niepowodzeń wielu projektów komputerowych spowodowanych tym, że projektanci nie wiedzieli, co będą tworzyć.

Przed przystąpieniem do projektowania bazy danych należy zapytać jej przyszłych użytkowników, czego od niej oczekują. Trzeba się dowiedzieć, jakich kwerend będą potrzebować oraz jakie będą drukować raporty.

W tym procesie zidentyfikuje się wymagania użytkowe — **szczególne** potrzeby docelowych użytkowników bazy danych. Należy zanotować uzyskane odpowiedzi i stworzyć uporządkowany zapis tego, jakie czynności powinna wykonywać gotowa baza danych. Pracując nad kolejnymi etapami cyklu życia projektu, trzeba wracać do tych wymagań i sprawdzać, czy projekt idzie we właściwym kierunku.

Projektowanie tabel

Tabele jako podmioty

Projektant bazy danych próbuje spojrzeć na pewien aspekt życia i zamodelować go w sposób zgodny z Accessem. Przykładowo, projekt bazy danych dla księgarni modeluje podmioty z życia, takie jak klienci księgarni oraz jej sprzedawcy, a także transakcje zawierane pomiędzy nimi (np. zakup książki lub przyjęcie zwrotu).

Aby zwizualizować proces projektowania, wyobraź sobie, że oglądasz film na temat fragmentu życia, o którym chcesz zapisać dane — na przykład o księgarni — a następnie zatrzymujesz go. Zwróć uwagę, jacy aktorzy biorą udział w scenie oraz jakie dane o nich warto zapisać.

Klient ma imię i adres — to będą początkowe dane w bazie. Dane te można wykorzystać do informowania klientów o specjalnych ofertach oraz aktualnych stanach magazynowych. Takie same dane należy zapisać o sprzedawcach pracujących w księgarni, tak by można było wysłać do nich przelew.

Warto również zapisać informacje o książkach w sprzedaży — na przykład cenę, tytuł i autora.

Wskazówka



Ponieważ tabele często modelują rzeczywiste obiekty lub pojęcia, nazwy tabel zawsze powinny być rzeczownikami, na przykład *Klient* lub *Samochód*.

Książki			
	Tytuł	Autor	Cena
	Co Cię Nie Zabije	Leokadia Martyniuk	39,99 zł
	Karol i Jan	Jerzy Szczeciński	21,99 zł
	Kot Na Cienkiej Blasze do Pieczenia	Artur Kowalski	21,99 zł
	Król Kniei	Leokadia Martyniuk	21,99 zł
	Miejsce Zwane Domem	Artur Kowalski	21,99 zł
	Podziwiał Marku!	Jerzy Szczeciński	21,99 zł
	Polne Ptaki	Artur Kowalski	21,99 zł
	Rzucanie Młotem - Wspomnienia	Krzysztof Filipiak	23,99 zł
	Szalona Składanka dla Dzieci	Jerzy Szczeciński	21,99 zł
	Zdumiewające Stworzenia	Artur Kowalski	21,99 zł

Wyobraź sobie scenę, którą chcesz zamodelować. Zapisz nazwiska aktorów oraz obiekty, które pojawiają się w ujęciu.

Nazwa tabeli powinna odpowiadać nazwie modelowanego podmiotu — na przykład *Klient*. Dane dotyczące podmiotu, na przykład *Nazwisko* oraz *Płeć*, będą polami projektowanej tabeli.

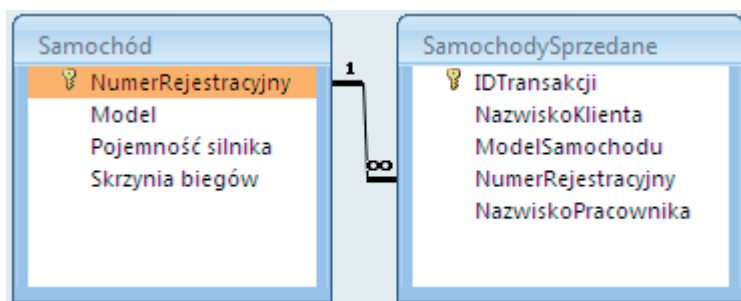
Klucze podstawowe i obce

Wybór kluczy podstawowych i obcych

Jak opisano na stronie 30., tabele są powiązane ze sobą za pomocą wspólnych pól. W jaki sposób wybiera się te pola?

W każdej tabeli musi być pole, które w unikatowy sposób identyfikuje indywidualne rekordy. Na przykład, gdybyśmy chcieli zidentyfikować każdy rekord w tabeli *Samochód*, moglibyśmy wykorzystać pole *NumerRejestracyjny*, ponieważ numer rejestracyjny jest unikatowym identyfikatorem każdego pojazdu. Z kolei gdybyśmy chcieli oznaczyć każdy rekord w tabeli *Pracownik*, moglibyśmy użyć pola *NumerNIP*, ponieważ numer NIP w swoisty sposób identyfikuje każdą osobę.

Pole, które w unikatowy sposób identyfikuje indywidualne rekordy w tabeli, określa się terminem **klucz podstawowy**. Przyjrzyjmy się polom, jakie wybraliśmy dla naszych tabel, i zobaczymy, które z nich może pełnić rolę klucza podstawowego. W tabeli może być tylko jedno pole pełniące funkcję klucza podstawowego. Jeżeli w tabeli znajduje się więcej takich pól, prawdopodobnie trzeba będzie podzielić tabelę na dwie lub więcej. W dalszej części niniejszego rozdziału wyjaśnimy dlaczego.



Definiowanie relacji pomiędzy dwoma tabelami polega na utworzeniu łącząca pomiędzy kluczem podstawowym a kluczem obcym. Klucz obcy to pole w tabeli, które dokładnie pasuje do klucza podstawowego innej tabeli. Na przykład, gdybyśmy chcieli zdefiniować relację pomiędzy wspomnianą wcześniej tabelą *Samochód* a inną tabelą o nazwie *SamochodySprzedane*, zawierającą rekordy wszystkich samochodów sprzedanych w określonej placówce, w tej drugiej tabeli należałoby umieścić pole *NumerRejestracyjny*, które pełniłoby rolę klucza obcego. Utworzenie relacji na podstawie dwóch pól *NumerRejestracyjny* pozwala na powiązanie informacji w obu tabelach w logiczny sposób, mający sens w rzeczywistości.

Nie zapomnij



W polu będącym kluczem podstawowym nie może być wartości null. Każdy rekord tabeli w tym polu musi zawierać jakąś wartość.

Strzeż się



Klucz podstawowy może się składać z dwóch lub więcej pól, które po połączeniu ze sobą w unikatowy sposób identyfikują poszczególne rekordy w tabeli. Taki sposób definiowania kluczy podstawowych nie jest jednak dobrym pomysłem i należy go unikać. Jedynym wyjątkiem od tej reguły jest sytuacja, w której trzeba utworzyć tabelę łączącą w celu zamodelowania relacji wiele do wielu.

Nie zapomnij



Klucz obcy w tabeli nie musi mieć takiej samej nazwy jak klucz podstawowy, z którym tworzy relację.

Nie zapomnij

Jeśli wydajność projektowanej bazy danych nie jest szczególnie istotna albo jeśli baza zawiera niewiele liczb bądź tabel, normalizowanie tabel do postaci normalnych wyższych niż pierwsza nie jest konieczne. Wykonanie tej czynności jest jednak zalecane.

Nie zapomnij

Jeśli tabele nie spełniają reguł drugiej i trzeciej postaci normalnej, należy je podzielić na dwie lub więcej tabel spełniających postacie normalne.

Doskonalenie projektu

Normalizacja

Ostatnią częścią procesu projektowania tabel jest **normalizacja**. Jest to proces stopniowego udoskonalania tabel w celu ochrony integralności i szczególności danych, które są w nich zapisane. Proces normalizacji umożliwia również zaoszczędzenie miejsca na dysku, ponieważ zmniejsza ryzyko redundancji danych (danych bezcelowo powtórzonych w innym miejscu). Normalizacja obejmuje restrukturyzację projektu do pierwszej, następnie drugiej, a na koniec trzeciej postaci normalnej.

Pierwsza postać normalna (1NF)

W tej postaci pole może zawierać tylko jedną wartość. Na przykład, adres należy podzielić na indywidualne pola. Nie wolno umieszczać całego adresu w jednym polu.

Druga postać normalna (2NF)

Każde pole w tabeli musi w całości zależeć od klucza podstawowego. Na przykład, w tabeli danych o pracownikach, w której kluczem podstawowym jest *NumerNIP* oraz która zawiera dwa pola — *Nazwisko* i *Oddział*, pole *Nazwisko* całkowicie zależy od pola *NumerNIP*. Jest tak dlatego, że pola *Nazwisko* i *NumerNIP* są ze sobą nierozdzielnie związane. Z kolei pole *Oddział* nie jest bezpośrednio związane z polem *NumerNIP*. Pracownik może pracować w dowolnym oddziale, ale zawsze będzie miał tylko jeden numer NIP.

Trzecia postać normalna (3NF)

W trzeciej postaci normalnej żadne z pól niebędących kluczem podstawowym nie może zależeć od innego pola, które nie jest kluczem podstawowym.

Definiowanie reguł biznesu

Decydowanie o tym, jakie tabele będą umieszczone w bazie danych oraz w jaki sposób będą ze sobą powiązane, to tylko jedna z części projektu. W celu jak najwierniejszego zamodelowania scenariuszy z życia należy przeanalizować reguły, które nimi rządzą. Reguły biznesu lub inaczej logika biznesowa to często nigdzie niezapisane i niepotwierdzone zasady postępowania, którymi kierujemy się na co dzień w pracy. Poza projektowaniem bazy danych nikt ich szczegółowo nie analizuje, ponieważ niejednokrotnie wynikają one ze zdrowego rozsądku.

Na przykład, jedna z reguł biznesu w banku brzmi: „limit debetu powinien wynosić 0,00 PLN lub mniej” albo „numer rachunku musi składać się z *n* cyfr”.

Należy zanotować reguły biznesu istotne dla bazy danych, tak by można było je zaimplementować i zapewnić ich przestrzeganie.