



DO NOWEJ
PODSTAWY PROGRAMOWEJ

Część 3

Tworzenie i administrowanie
bazami danych

Kwalifikacja EE.09

Programowanie, tworzenie
i administrowanie stronami
internetowymi i bazami danych



Podręcznik do nauki zawodu
technik informatyk

Jolanta Pokorska

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Wydawnictwo HELION dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Wydawnictwo HELION nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Redaktor prowadzący: Joanna Zaręba

Projekt okładki: Jan Paluch

Fotografia na okładce została wykorzystana za zgodą Shutterstock.

Wydawnictwo HELION

ul. Kościuszki 1c, 44-100 GLIWICE

tel. 32 231 22 19, 32 230 98 63

e-mail: helion@helion.pl

WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie?e093ti>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzje.

ISBN: 978-83-283-4863-9

Copyright © Helion 2019

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

Spis treści

Wstęp	5
Rozdział 1. Zasady projektowania relacyjnych baz danych	7
1.1. Wprowadzenie do baz danych	7
1.2. Modele baz danych	8
1.3. Relacyjny model danych	11
1.4. Projektowanie bazy danych	18
1.5. Cechy relacyjnej bazy danych	46
1.6. Pytania i zadania	49
Rozdział 2. Tworzenie lokalnych baz danych w programie MS Access	51
2.1. Wprowadzenie	51
2.2. Środowisko pracy programu Access	52
2.3. Tabela jako podstawowa forma organizacji danych	53
2.4. Definiowanie wyrażeń w Accessie	68
2.5. Kwerendy	73
2.6. Formularze	87
2.7. Raporty	112
2.8. Moduły	116
2.9. Pytania i zadania	116
Rozdział 3. Systemy zarządzania bazą danych	119
3.1. Wprowadzenie	119
3.2. Serwery bazodanowe	122
3.3. Pytania i zadania	136
Rozdział 4. SQL — strukturalny język zapytań	139
4.1. Wprowadzenie	139
4.2. Standardy języka SQL	139
4.3. Składnia języka SQL	140
4.4. Język definiowania danych (DDL)	145
4.5. Język manipulowania danymi (DML)	160

4.6. Instrukcja SELECT	164
4.7. Łączenie tabel	176
4.8. Więzy integralności (MS SQL)	182
4.9. Łączenie wyników zapytań	187
4.10. Podzapytania	189
4.11. Widoki (perspektywy)	196
4.12. Indeksy	198
4.13. Transakcje	199
4.14. Współbieżność	204
4.15. T-SQL	211
4.16. Pytania i zadania	224
Rozdział 5. Administrowanie serwerami baz danych	233
5.1. Wprowadzenie	233
5.2. MS SQL Server	234
5.3. Prawa dostępu do serwera	240
5.4. Kopie bezpieczeństwa	246
5.5. Import i eksport danych	250
5.6. MySQL	253
5.7. Prawa dostępu do serwera	256
5.8. Kopie bezpieczeństwa	261
5.9. Import i eksport danych	264
5.10. Optymalizacja wydajności SZBD	266
5.11. Pytania i zadania	271
Bibliografia	273
Skorowidz	274

Wstęp

Kwalifikacja EE.09. Programowanie, tworzenie i administrowanie stronami internetowymi i bazami danych. Część 3. Tworzenie i administrowanie bazami danych. Podręcznik do nauki zawodu technik informatyk jest trzecim z grupy podręczników przygotowanych dla kwalifikacji EE.09. Programowanie, tworzenie i administrowanie stronami internetowymi i bazami danych. Kwalifikacja ta, wraz z kwalifikacją EE.08. Montaż i eksploatacja systemów komputerowych, urządzeń peryferyjnych i sieci, obejmuje efekty kształcenia, których osiągnięcie jest niezbędne do uzyskania dyplomu potwierdzającego kwalifikacje zawodowe w zawodzie technik informatyk.

Treści zawarte w trzeciej części podręcznika zostały oparte na podstawie programowej kształcenia w zawodach wprowadzonej rozporządzeniem Ministra Edukacji Narodowej z dnia 31 marca 2017 r.

Obejmują one zagadnienia teoretyczne prowadzące do uzyskania wymienionych w podstawie programowej efektów kształcenia, projekty różnych zadań oraz ich praktyczną realizację. Tak skonstruowany podręcznik pomaga uczniowi w zdobywaniu wymaganej wiedzy oraz umożliwia mu samodzielne poszerzanie umiejętności.

Podręcznik składa się z pięciu rozdziałów. Ich budowa pozwala na realizację treści programowych w sposób wybrany przez nauczyciela.

Rozdział 1., „Zasady projektowania relacyjnych baz danych”, zawiera omówienie podstawowych zagadnień związanych z relacyjnymi bazami danych. Dotyczy efektów związanych z umiejętnością posługiwania się podstawowymi pojęciami z zakresu relacyjnych baz danych oraz stosowaniem relacyjnych baz danych. Efektami tymi są: identyfikowanie podstawowych pojęć dotyczących relacyjnych baz danych, identyfikowanie elementów bazy danych, stosowanie zasad projektowania baz danych, tworzenie graficznych schematów baz danych, projektowanie tabel baz danych, identyfikowanie i stosowanie zasad normalizacji tabel, definiowanie związków między tabelami bazy danych, identyfikowanie właściwości relacyjnych baz danych.

Rozdział 2., „Tworzenie lokalnych baz danych w programie MS Access”, zawiera omówienie zagadnień związanych z tworzeniem bazy danych, tworzeniem obiektów bazy danych oraz zarządzaniem bazą danych w programie MS Access. Dotyczy efektów związanych z tworzeniem tabel, formularzy, zapytań i raportów oraz modyfikowaniem i rozbudową struktury bazy danych. Efektami tymi są: projektowanie formularzy do przetwarzania danych, wykorzystywanie podformularzy do przetwarzania danych, identyfikowanie i stosowanie różnych rodzajów kwerend do przetwarzania danych, projektowanie i wykonywanie raportów z bazy danych, modyfikowanie obiektów oraz struktury bazy danych, rozbudowywanie struktury bazy danych.

Rozdział 3., „Systemy zarządzania bazą danych”, zawiera omówienie zagadnień związanych z architekturą systemów baz danych oraz z instalowaniem i konfigurowaniem systemów bazodanowych i systemów zarządzania bazą danych. Dotyczy efektów związanych z instalowaniem systemów baz danych, konfigurowaniem serwerów baz danych, wykorzystaniem narzędzi graficznych do tworzenia baz danych oraz do zarządzania bazami. Efektami tymi są: identyfikowanie różnych systemów zarządzania bazami danych, identyfikowanie funkcji systemów zarządzania bazami danych, identyfikowanie serwerów baz danych, instalowanie systemów baz danych, zarządzanie bazami danych, stosowanie narzędzi graficznych do tworzenia baz danych.

Rozdział 4., „SQL — strukturalny język zapytań”, zawiera omówienie zagadnień związanych z tworzeniem bazy danych i zarządzaniem nią w języku SQL. Dotyczy efektów związanych z korzystaniem z funkcji strukturalnego języka zapytań oraz z posługiwaniem się strukturalnym językiem zapytań do obsługi baz danych. Efektami tymi są: identyfikowanie i stosowanie składni strukturalnego języka zapytań, stosowanie funkcji strukturalnego języka zapytań, stosowanie instrukcji strukturalnego języka zapytań dotyczących tworzenia struktury bazy danych, tworzenie zapytań, podzapytań i połączeń między tabelami, identyfikowanie i stosowanie transakcji przy użyciu strukturalnego języka zapytań, stosowanie instrukcji języka T-SQL.

Rozdział 5., „Administrowanie serwerami baz danych”, zawiera omówienie zagadnień związanych z administrowaniem serwerami bazodanowymi. Dotyczy efektów związanych z określaniem uprawnień poszczególnych użytkowników, określaniem zabezpieczeń dla użytkowników, dobieraniem sposobów ustawienia zabezpieczeń dostępu do danych, zarządzaniem kopiami zapasowymi baz danych i ich odzyskiwaniem, kontrolowaniem spójności bazy danych, dokonywaniem naprawy bazy danych. Efektami tymi są: zarządzanie bazą danych, kontrolowanie spójności bazy danych, identyfikowanie i dobieranie sposobów ustawienia zabezpieczeń dostępu do danych, zarządzanie bezpieczeństwem bazy danych, identyfikowanie uprawnień użytkowników bazy danych, zarządzanie kopiami zapasowymi baz danych, zarządzanie odzyskiwaniem danych.

1.4.4. Tworzenie modelu konceptualnego (diagramy ERD)

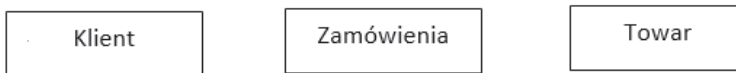
Konceptualne projektowanie bazy danych to konstruowanie schematu danych niezależnego od wybranego modelu danych, docelowego systemu zarządzania bazą danych, programów użytkowych czy języka programowania.

Do tworzenia modelu graficznego schematu bazy danych wykorzystywane są diagramy związków encji, z których najpopularniejsze są diagramy ERD (ang. *Entity Relationship Diagram*). Pozwalają one na modelowanie struktur danych oraz związków zachodzących między tymi strukturami. Nadają się szczególnie do modelowania relacyjnych baz danych, ponieważ umożliwiają prawie bezpośrednie przekształcenie diagramu w schemat relacyjny. Pozwalają na analizę struktury bazy danych, mogą też stanowić część dokumentacji tworzonego systemu baz danych.

Na diagramy ERD składają się trzy rodzaje elementów:

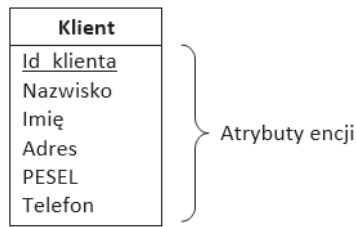
- zbiory encji,
- atrybuty encji,
- związki zachodzące między encjami.

Encja to reprezentacja obiektu przechowywanego w bazie danych. Graficzną reprezentacją encji jest najczęściej prostokąt (rysunek 1.8).

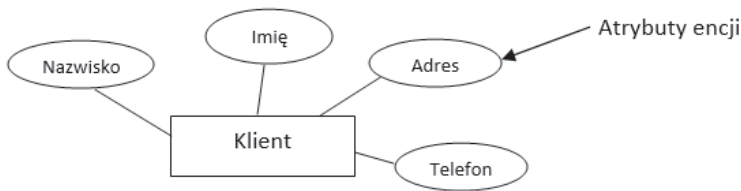


Rysunek 1.8. Graficzna reprezentacja encji

Atrybut opisuje encję. Może on być liczbą, tekstem lub wartością logiczną. W relacyjnym modelu baz danych atrybut jest reprezentowany przez kolumnę tabeli. Graficzna reprezentacja atrybutów dla encji *Klient* została pokazana na rysunkach 1.9 i 1.10.



Rysunek 1.9. Atrybuty encji Klient

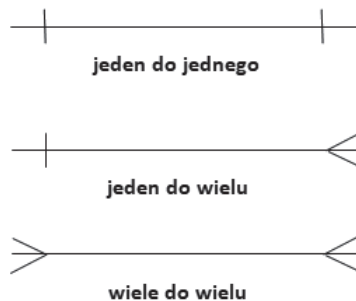


Rysunek 1.10. Atrybuty encji Klient

Związek to powiązanie między dwoma zbiorami encji. Każdy związek ma dwa końce, do których są przypisane następujące atrybuty:

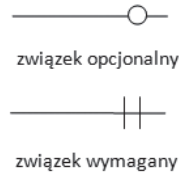
- nazwa,
- stopień związku,
- uczestnictwo lub opcjonalność związku.

Stopień związku określa, jakiego typu związek zachodzi między encjami. „Jeden do jednego” oznacza, że encji odpowiada dokładnie jedna encja. „Jeden do wielu” oznacza, że encji odpowiada jedna lub kilka encji. „Wiele do wielu” oznacza, że encji lub kilku encjom odpowiada jedna lub kilka encji. Opis reprezentacji graficznej stopnia związku został pokazany na rysunku 1.11.



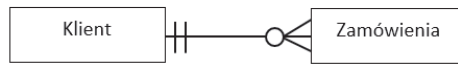
Rysunek 1.11. Graficzna reprezentacja związków zachodzących między encjami

Opcjonalność związku określa, czy związek jest opcjonalny, czy wymagany. Opis reprezentacji graficznej opcjonalności związku został pokazany na rysunku 1.12.



Rysunek 1.12. Graficzna reprezentacja opcjonalności związku

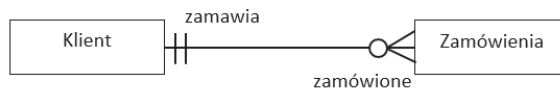
Przykład prostego diagramu związków encji został pokazany na rysunku 1.13. *Zamówienie* musi mieć przypisanego *Klienta*, natomiast *Klient* może złożyć *Zamówienie*. *Klient* może złożyć wiele *Zamówień*, ale złożone *Zamówienie* dotyczy tylko jednego *Klienta*.



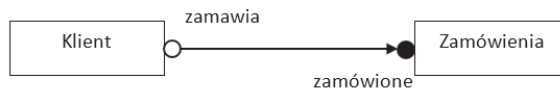
Rysunek 1.13. Diagram związków encji

Diagramy ERD spotyka się w wielu różnych notacjach, na przykład: Martina, Bachmana, Chena, IDEF1X (rysunek 1.14).

Notacja Martina



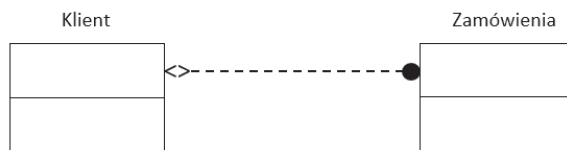
Notacja Bachmana



Notacja Chena



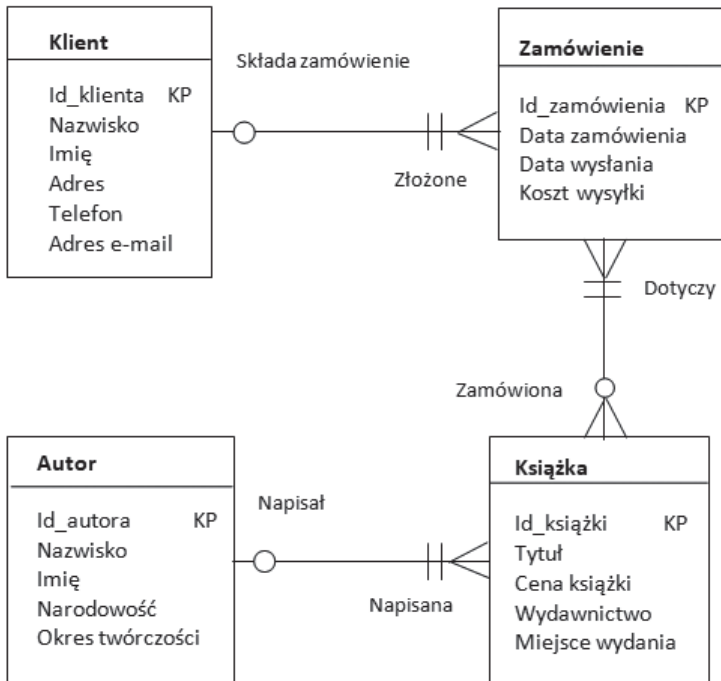
Notacja IDEF1X



Rysunek 1.14. Diagram związków encji zapisany w różnych notacjach

Istnieje wiele narzędzi wspomagających rysowanie diagramów ERD, ale jedynie w przypadku narzędzi klasy CASE (ang. *Computer Aided Software Engineering*) można mówić o określonej notacji. Narzędzia CASE są wykorzystywane do projektowania oprogramowania. Pozwalają tworzyć model graficzny oraz poprzez generowanie gotowych skryptów wspomagają wytwarzanie oprogramowania.

Prosty przykład diagramu ERD w notacji Martina dla księgarni został przedstawiony na rysunku 1.15.



Rysunek 1.15. Diagram ERD w notacji Martina dla księgarni

W pokazanym na rysunku schemacie encje zostały przedstawione za pomocą prostokątów zawierających listę atrybutów. Klucze główne zostały oznaczone symbolem KP. Stopień związku i uczestnictwo zostały oznaczone liniami łączącymi z odpowiednimi symbolami opisującymi stopień oraz opcjonalność związku. Należy zwrócić uwagę na to, że w encjach nie umieszcza się kluczy obcych. Zostaną one dodane na etapie przekształcania encji w tabelę.

Tak przygotowany diagram ERD pozwala na późniejszą weryfikację i optymalizację bazy danych, a także stanowi podstawową dokumentację projektowanej bazy danych. Można go również wykorzystać w jednym z narzędzi CASE do wygenerowania fizycznej struktury bazy danych.

1.4.5. Projektowanie bazy danych za pomocą narzędzi CASE

Narzędzia CASE (ang. *Computer Aided Software Engineering*) są wykorzystywane podczas projektowania różnego rodzaju oprogramowania, najczęściej wspomagają proces jego wytwarzania.

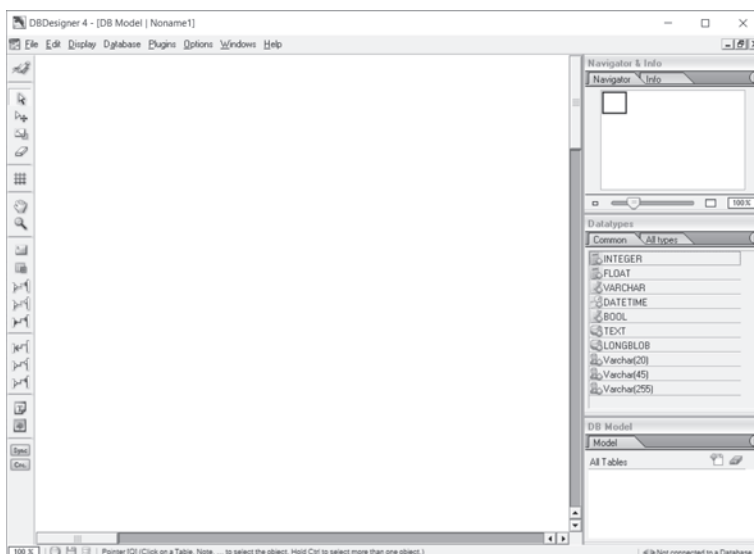
Narzędzia te pozwalają tworzyć modele graficzne odpowiadające konstrukcjom programistycznym. Wykorzystywane są tutaj edytory notacji graficznych, które dają możliwości tworzenia diagramów i powiązań między poszczególnymi elementami. Bardziej zaawansowane edytory umożliwiają przetwarzanie informacji i udostępnianie danych do aplikacji zewnętrznych, na przykład kodów w językach Visual Basic, SQL, ODBC.

Narzędzia CASE mogą być stosowane do generowania kodu na podstawie zaprojektowanego modelu danych, można również za ich pomocą, na podstawie analizy kodu źródłowego, odtworzyć projekt i specyfikację bazy danych.

Przykładem narzędzia typu CASE jest program DBDesigner4. Jest to narzędzie do wizualnego projektowania, modelowania i tworzenia baz danych. Program został stworzony z myślą o bazie MySQL, ale obsługuje również bazy danych Oracle, SQLite, MS SQL. Jest rozpowszechniany jako open source i jest dostępny na stronie <http://fabforce.net/index.php>.

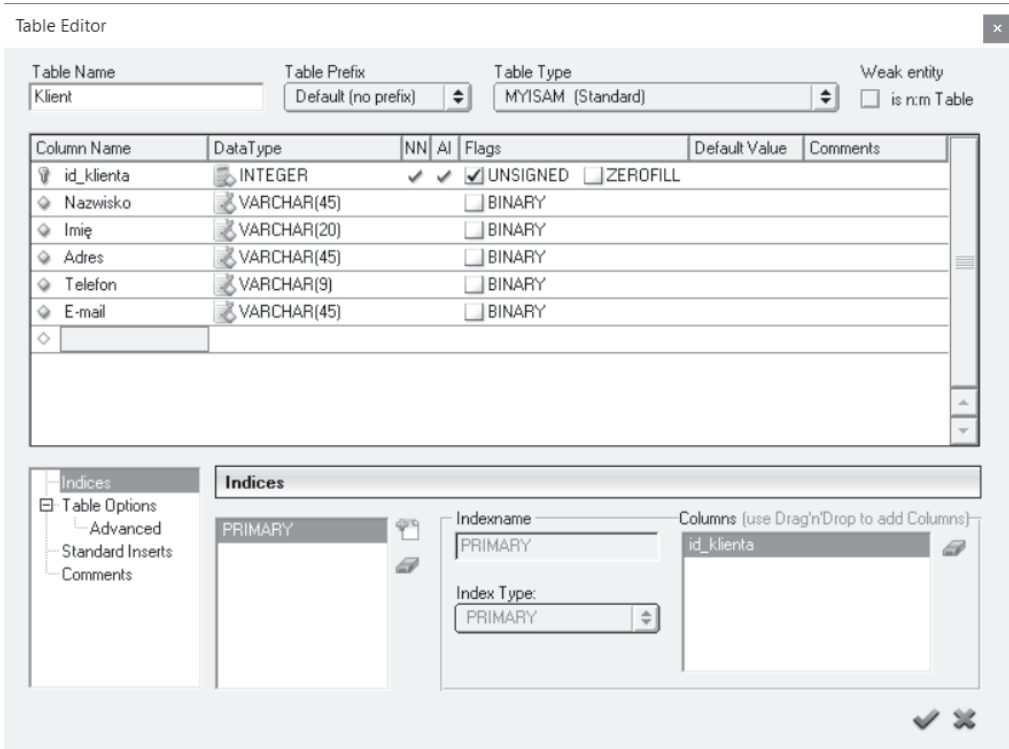
Zainstalowany program można wykorzystać do przygotowania logicznego projektu bazy danych.

Okno programu DBDesigner4 składa się z pięciu obszarów (rysunek 1.16). Pusty obszar na środku ekranu to obszar roboczy. Z lewej strony znajduje się pasek narzędzi. Z prawej w górnej części znajduje się okno nawigacji i informacji, na środku okno typów danych, a w dolnej części okno bieżącego modelu bazy danych.



Rysunek 1.16. Okno programu DBDesigner4

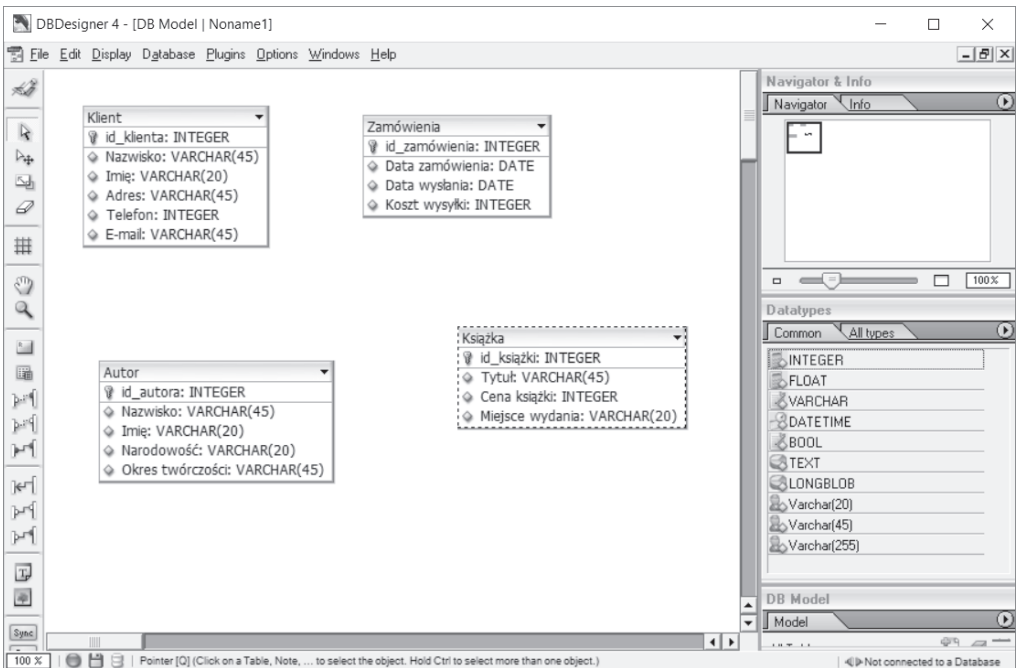
Pracę w programie rozpoczynamy od utworzenia nowego projektu (menu *File/New*). Kolejnym etapem jest utworzenie pierwszej tabeli. Należy wybrać na pasku narzędzi ikonę tworzenia tabeli (*New Table*) i kliknąć obszar roboczy. W efekcie zostanie utworzona tabela. Po dwukrotnym kliknięciu tabeli można otworzyć okno jej edytowania, w którym należy wprowadzić nazwę tabeli i nazwy pól oraz wybrać ich typy (rysunek 1.17).



Rysunek 1.17. Okno edytowania tabeli

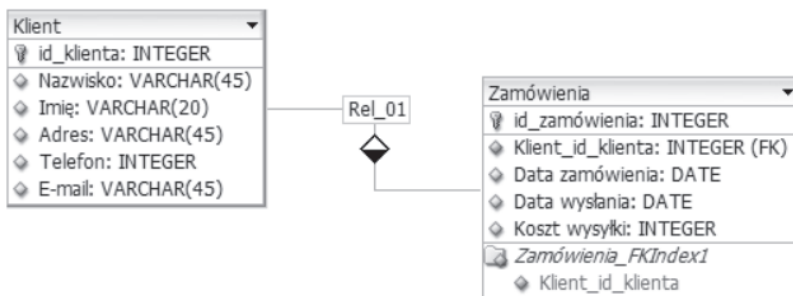
W oknie edytowania tabeli w kolumnie *Column Name* należy wprowadzić nazwę tworzonego pola, w kolumnie *Data Type* określić typ danych, w kolumnie *NN* określić, czy dozwolona jest wartość *NULL (NOT NULL)*, w kolumnie *AI* zaznaczyć automatyczne zwiększanie wartości o 1 (*AUTO INCREMENT*), w kolumnie *Flags* zdefiniować dodatkowe opcje zależne od typu danych, w kolumnie *Default Value* ustawić wartość domyślną pola, a w kolumnie *Comments* wstawić komentarz.

Dla projektowanego modelu graficznego bazy danych w podobny sposób należy utworzyć pozostałe tabele (rysunek 1.18).



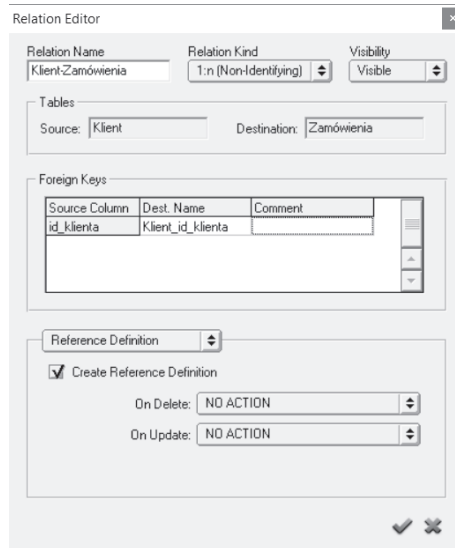
Rysunek 1.18. Tabele zaprojektowane w programie DBDesigner4

Po utworzeniu wszystkich tabel należy zdefiniować połączenia między nimi. Program DBDesigner4 obsługuje wszystkie rodzaje połączeń występujących w bazie danych. Ikony odpowiednich połączeń są dostępne na pasku narzędzi. Aby dodać połączenie typu „jeden do wielu”, należy wybrać ikonę *New 1:n*. Po wybraniu ikony rodzaju połączenia klikamy najpierw tabelę ze strony „jeden” (*Klient*), a następnie tabelę ze strony „wiele” (*Zamówienia*). W wyniku zdefiniowania połączenia w tabeli ze strony „wiele” (*Zamówienia*) pojawiło się nowe pole (*Klient_id_klienta*), opisujące związek między tabelami, które stanie się kluczem obcym (rysunek 1.19).



Rysunek 1.19. Definiowanie połączenia jeden do wielu

Aby edytować utworzone połączenie, należy dwukrotnie kliknąć narysowaną linię. Zostanie otwarte okno edytowania połączenia, w którym można zmienić nazwę relacji oraz nazwę pola klucza obcego (rysunek 1.20).



Rysunek 1.20. Okno edytowania połączenia

W podobny sposób należy zdefiniować połączenie między tabelami *Autor* i *Książka*.

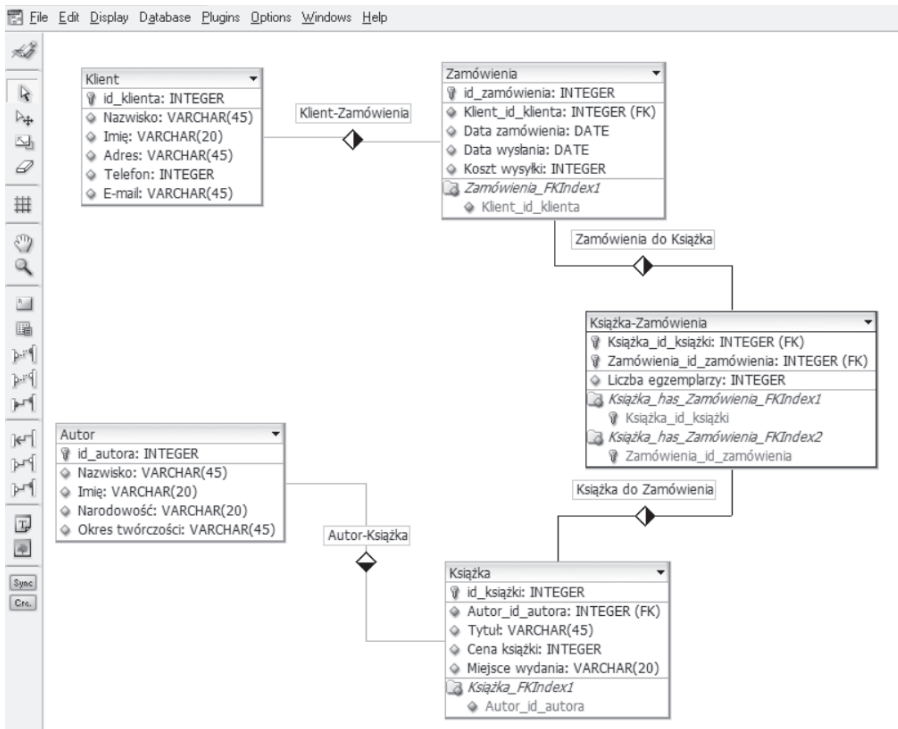
Ponieważ w jednym zamówieniu może znaleźć się kilka książek, a jedna książka może zostać wskazana w wielu zamówieniach, między tabelami *Zamówienia* i *Książki* występuje połączenie „wiele do wielu”. Aby dodać taki typ połączenia, należy wybrać ikonę *New n:m*. Po wybraniu tej ikony klikamy na przykład tabelę *Książka*, a następnie tabelę *Zamówienia* (lub w odwrotnej kolejności). W wyniku zdefiniowania połączenia w projekcie zostanie umieszczona dodatkowa tabela opisująca zdefiniowane połączenie. Będzie ona zawierała klucze podstawowe łączonych tabel (rysunek 1.21).



Rysunek 1.21. Definiowanie połączenia „wiele do wielu”

Do utworzonej tabeli można dodać pola będące atrybutami połączenia, np. gdy chcemy przechowywać informację, w ilu egzemplarzach została zamówiona każda książka.

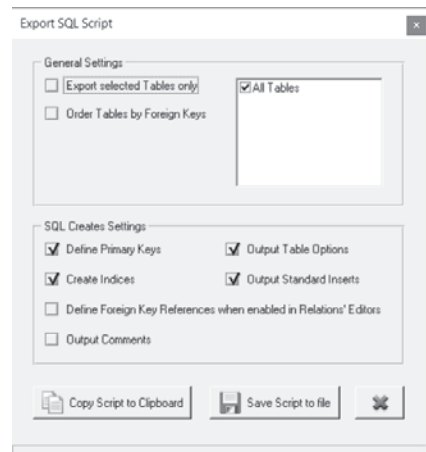
Po utworzeniu połączeń i dodaniu potrzebnych atrybutów uzyskamy efekt podobny do pokazanego na rysunku 1.22.



Rysunek 1.22. Schemat bazy danych uzyskany w programie DBDesigner4

Utworzony projekt należy zapisać w pliku, wybierając z menu *File/Save*. Można również zaprojektowaną bazę danych wyeksportować do pliku *.sql*. W tym celu należy wybrać z menu *File/Export/SQL Create Script* i w otwartym oknie zaznaczyć opcje, tak jak pokazano na rysunku 1.23, a następnie kliknąć przycisk *Save Script to file*.

Po wykonaniu tych czynności zostanie wygenerowany skrypt, którego zawartość można zobaczyć, otwierając plik na przykład w edytorze tekstowym Notepad++ (rysunek 1.24).



Rysunek 1.23. Opcje eksportowania projektu bazy do kodu SQL

```

1 CREATE TABLE Autor (
2     id_autora INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
3     Nazwisko VARCHAR(45) NULL,
4     Imię VARCHAR(20) NULL,
5     Narodowość VARCHAR(20) NULL,
6     Okres twórczości VARCHAR(45) NULL,
7     PRIMARY KEY(id_autora)
8 );
9
10 CREATE TABLE Klient (
11     id_klienta INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
12     Nazwisko VARCHAR(45) NULL,
13     Imię VARCHAR(20) NULL,
14     Adres VARCHAR(45) NULL,
15     Telefon INTEGER UNSIGNED NULL,
16     E-mail VARCHAR(45) NULL,
17     PRIMARY KEY(id_klienta)
18 );
19
20 CREATE TABLE Książka (
21     id_książki INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
22     Autor_id_autora INTEGER UNSIGNED NOT NULL,
23     Tytuł VARCHAR(45) NULL,
24     Cena książki INTEGER UNSIGNED NULL,
25     Miejsce wydania VARCHAR(20) NULL,
26     PRIMARY KEY(id_książki),
27     INDEX Książka_FKIndex1(Autor_id_autora)
28 );
29
30 CREATE TABLE Książka-Zamówienia (
31     Książka_id_książki INTEGER UNSIGNED NOT NULL,
32     Zamówienia_id_zamówienia INTEGER UNSIGNED NOT NULL,
33     Liczba egzemplarzy INTEGER UNSIGNED NULL,
34     PRIMARY KEY(Książka_id_książki, Zamówienia_id_zamówienia),
35     INDEX Książka_has_Zamówienia_FKIndex1(Książka_id_książki),
36     INDEX Książka_has_Zamówienia_FKIndex2(Zamówienia_id_zamówienia)
37 );
38
39 CREATE TABLE Zamówienia (
40     id_zamówienia INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
41     Klient_id_klienta INTEGER UNSIGNED NOT NULL,
42     Data zamówienia DATE NULL,

```

Rysunek 1.24. Kod SQL wygenerowany automatycznie za pomocą programu DBDesigner4

Tak przygotowany skrypt może zostać uruchomiony na jednym z serwerów bazodanych w celu wygenerowania gotowej bazy danych.

1.4.6. Transformacja modelu konceptualnego do modelu relacyjnego

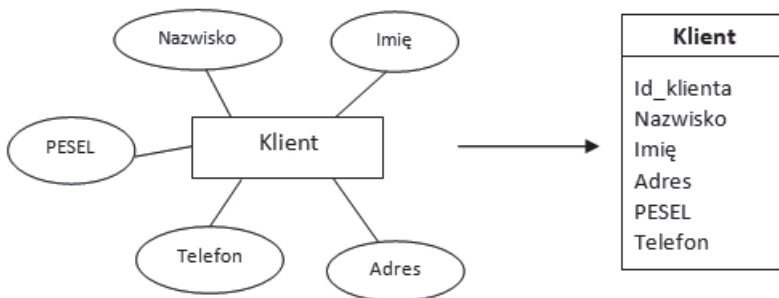
Jeżeli w projektowanej bazie danych występuje kilka zbiorów encji (na przykład encje *Książka* i *Autor*), między tymi encjami zachodzą związki. Każda książka jest dziełem określonego autora, każdy autor ma przypisany zbiór książek. Zarówno zbiory encji, jak i związki zachodzące między encjami zapisujemy w tabelach projektowanej bazy danych. Przy projektowaniu tabel stosujemy reguły, które określają, w jaki sposób należy przekształcać model bazy danych utworzony za pomocą diagramów ERD i narzędzi CASE w schemat bazy danych.

Reguły transformacji diagramów ERD do tabel

Transformacja encji do tabeli

WSKAZÓWKA

Do opisu każdego zbioru podobnych encji stosuje się oddzielną tabelę (rysunek 1.25). Jednej encji odpowiada jeden wiersz. Atrybutowi odpowiada kolumna. Dla każdego atrybutu określa się typ informacji.

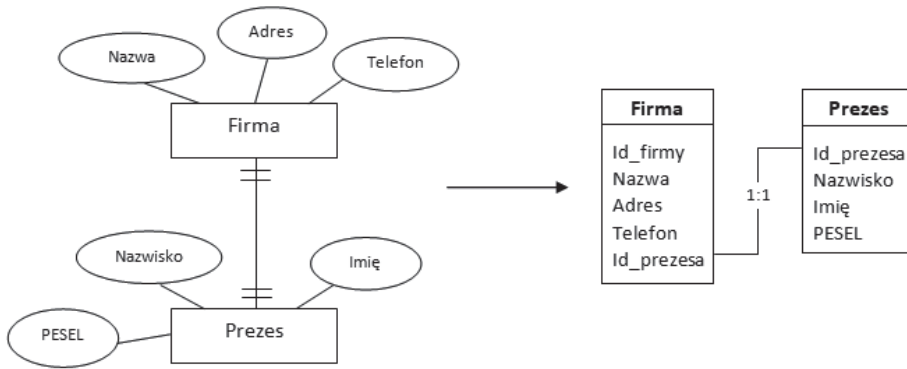


Rysunek 1.25. Transformacja encji do tabeli

Transformacja związku „jeden do jednego”

WSKAZÓWKA

Zapis związku „jeden do jednego” może zostać umieszczony w dodatkowej kolumnie w tabeli należącej do związku. Kolumna ta może znaleźć się w dowolnej tabeli. Dołączona kolumna zawiera klucz tabeli, z którą zachodzi związek (rysunek 1.26).

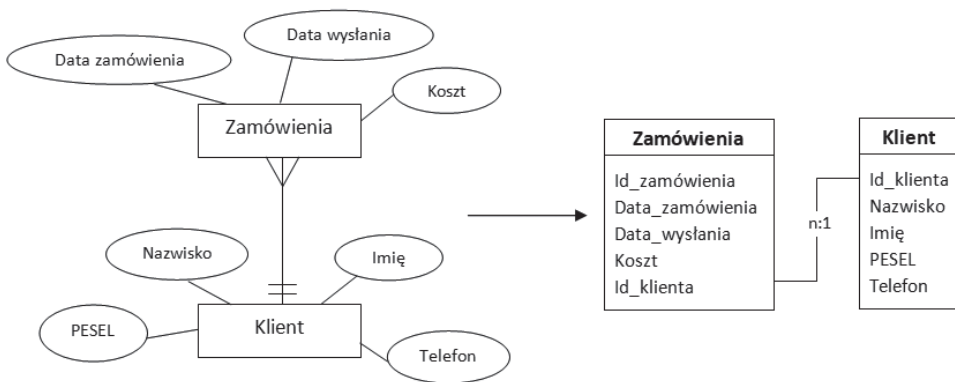


Rysunek 1.26. Transformacja związku „jeden do jednego” do tabeli

Transformacja związku „wiele do jednego”

WSKAZÓWKA

Zapis związku „wiele do jednego” może być umieszczony w dodatkowej kolumnie w tabeli należącej do związku. Kolumna ta musi znaleźć się w tabeli ze strony „wiele”. Dołączona kolumna zawiera klucz tabeli, z którą zachodzi związek (rysunek 1.27).

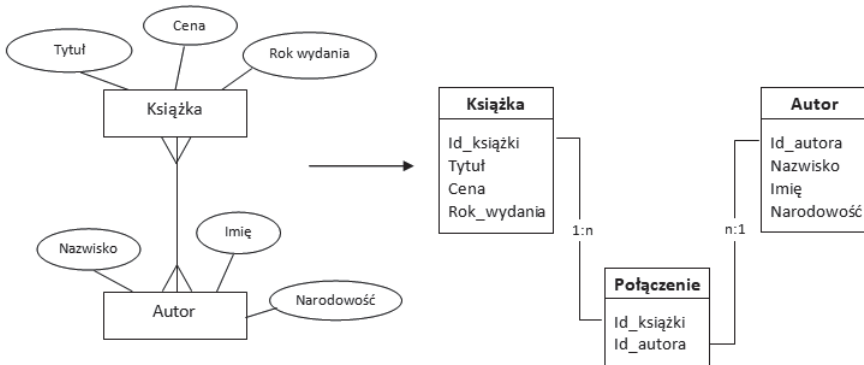


Rysunek 1.27. Transformacja związku „wiele do jednego” do tabeli

Transformacja związku „wiele do wielu”

WSKAZÓWKA

Związek „wiele do wielu” opisuje się w oddzielnej tabeli, której kolumny tworzone są z kluczy encji należących do związku (rysunek 1.28).

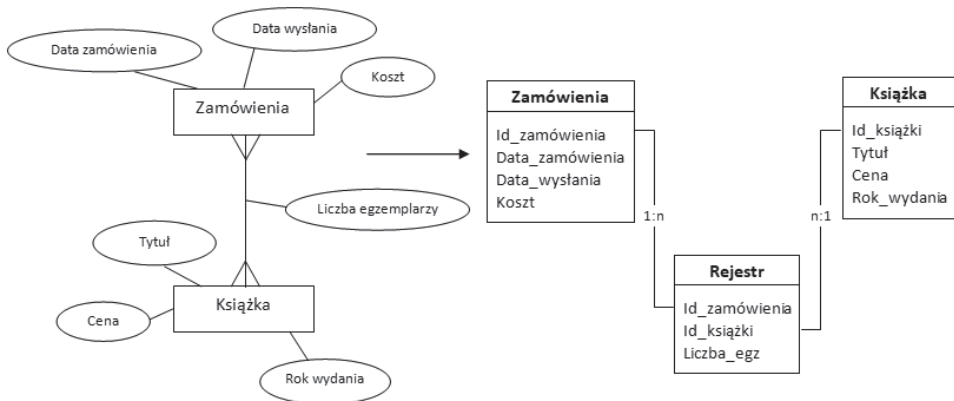


Rysunek 1.28. Transformacja związku „wiele do wielu” do tabeli

Transformacja związku „wiele do wielu” z atrybutami

WSKAZÓWKA

Jeżeli związek „wiele do wielu” posiada atrybuty, należy je umieścić w tabeli opisującej związek (rysunek 1.29).



Rysunek 1.29. Transformacja związku „wiele do wielu” z atrybutami

Jeśli klucze w tabeli opisującej związek składają się z wielu atrybutów lub są długie, należy zastąpić je kluczami sztucznymi.

1.4.7. Normalizacja tabel

Normalizację stosuje się, aby sprawdzić, czy zaprojektowane tabele mają prawidłową strukturę. Proces normalizacji rozpoczynamy, gdy zostanie utworzony wstępny projekt tabel. Pozwala ona określić, czy informacje przewidziane w projekcie bazy zostały przydzielone do właściwych tabel. Natomiast nie da odpowiedzi na pytanie, czy projekt bazy danych jest prawidłowy.

Korzyści płynące z normalizacji tabel są następujące:

- zlikwidowanie problemu powtarzania danych,
- optymalizacja objętości bazy danych,
- optymalizacja efektywności obsługi bazy danych,
- minimalizacja zagrożenia błędami przy wprowadzaniu danych.

Normalizacja bazy danych wymaga rozbicia dużych tabel na mniejsze. Zmniejsza to wydajność bazy, dlatego w niektórych przypadkach nie normalizuje się tabel — szczególnie w systemach niekorzystających z modelu relacyjnego.

Stosowane są cztery reguły normalizacji, ale w większości projektów baz danych wystarczy sprawdzić trzy pierwsze. Zostaną one omówione poniżej.

Dla każdej z nich stosowane są określenia: pierwsza postać normalna (**I PN**), druga postać normalna (**II PN**) i trzecia postać normalna (**III PN**).

Pierwsza postać normalna

DEFINICJA

Tabela jest w pierwszej postaci normalnej (**I PN**), gdy każdy wiersz w tabeli przechowuje informacje o pojedynczym obiekcie, a każde pole tabeli zawiera informację elementarną (atomową).

Oznacza to, że w komórce tabeli nie może wystąpić lista wartości, na przykład w polu *Narodowość autora* nie można umieścić dwóch narodowości — polskiej i angielskiej.

Załóżmy, że w projektowanej bazie danych dla księgarni została zaprojektowana tabela *Realizacja zamówień* z polami: *Nazwisko klienta*, *Imię*, *Adres*, *Telefon*, *PESEL*, *Tytuł książki*, *Liczba egzemplarzy*, *Cena*. W polu *Tytuł książki* będą umieszczane tytuły książek zakupionych przez klienta. Gdy klient kupi dwie książki, w polu *Tytuł książki* należałoby wpisać dwa tytuły. Powstałaby lista wartości (rysunek 1.30). Tak zaprojektowana tabela nie jest w **I PN**. Nie będzie możliwe prawidłowe przetwarzanie danych zapisanych w tabeli. Rozwiązaniem jest zapisanie informacji o zakupionych książkach

w dwóch wierszach. W pierwszym wierszu w polu *Tytuł książki* należy wpisać tytuł pierwszej książki, w drugim — tytuł drugiej książki, natomiast nazwisko klienta zostanie powtórzone w liczbie wierszy równej liczbie zakupionych książek (rysunek 1.31). Teraz tabela jest w **I PN**.

Realizacja zamówień							
Nazwisko kli	Imię	Adres	Telefon	PESEL	Tytuł książki	Liczł	Cena
Nowak	Marek	Toruń	(56) 6589234	79120307431	Dziady	2	20,00 zł
Kowalski	Adam	Warszawa	(22) 3451234	80122401871	Balladyna, Tango	1	15,00 zł
Górecki	Grzegorz	Poznań	(45) 2367897	82061203983	Pan Tadeusz	1	21,00 zł
Zan	Marcin	Gdańsk	(33) 8373635	82020201875			

Rysunek 1.30. Tabela nie jest w **I PN**, ponieważ w polu Tytuł książki pojawia się lista wartości

Realizacja zamówień							
Nazwisko klie	Imię	Adres	Telefon	PESEL	Tytuł książki	Liczł	Cena
Nowak	Marek	Toruń	(56) 6589234	79120307431	Dziady	2	20,00 zł
Kowalski	Adam	Warszawa	(22) 3451234	80122401871	Balladyna	1	15,00 zł
Kowalski	Adam	Warszawa	(22) 3451234	80122401871	Tango	2	18,00 zł
Górecki	Grzegorz	Poznań	(45) 2367897	82061203983	Pan Tadeusz	1	21,00 zł

Rysunek 1.31. Tabela jest w **I PN**, ponieważ w polu Tytuł książki występują pojedyncze wartości

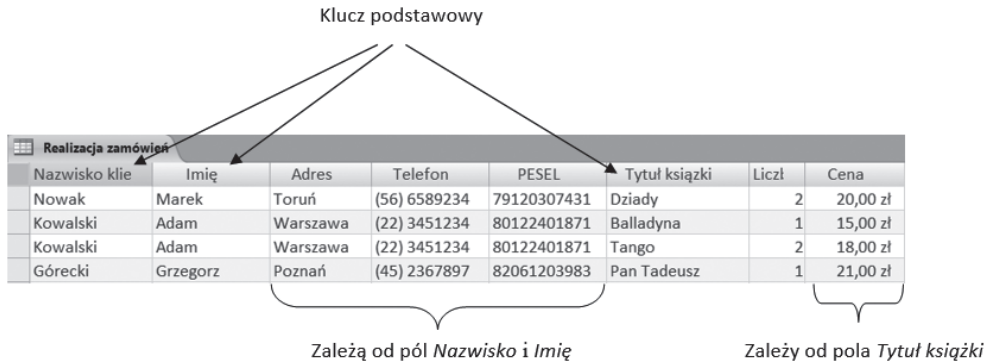
Druga postać normalna

DEFINICJA

Tabela jest w drugiej postaci normalnej (**II PN**), jeżeli jest w pierwszej postaci normalnej (**I PN**) oraz każde z pól niewchodzących w skład klucza podstawowego zależy od całego klucza, a nie od jego części.

Ta reguła i następna służą do sprawdzenia, czy w tabeli i bazie danych nie doszło do redundancji, czyli niepotrzebnego powtarzania danych. Ponieważ **II PN** odnosi się do klucza podstawowego, należy określić ten klucz dla tabeli. Jeżeli klucz podstawowy składa się z jednego pola, tabela jest w **II PN**, ponieważ wszystkie pola, poza polem klucza podstawowego, muszą odnosić się do pola klucza podstawowego.

W zaprojektowanej tabeli *Realizacja zamówień* kluczem podstawowym będzie kombinacja pól *Nazwisko klienta*, *Imię* oraz *Tytuł książki*. Pola niewchodzące w skład klucza podstawowego (*Adres*, *Telefon*, *PESEL*) zależą od pól *Nazwisko* i *Imię*, natomiast nie zależą od pola *Tytuł książki*. Pole *Cena* zależy jedynie od pola *Tytuł książki* (rysunek 1.32). Tylko pole *Liczba egzemplarzy* zależy zarówno od pól *Nazwisko* i *Imię*, jak i od pola *Tytuł książki*. Tabela nie jest w **II PN**.



Rysunek 1.32. Tabela nie jest w II PN, ponieważ tylko pole Liczba egzemplarzy zależy od całego klucza

Normalizacja polega na podzieleniu tabeli na takie tabele, które spełnią warunek II PN. Tabelę *Realizacja zamówień* należy podzielić na trzy tabele: *Klient*, z polami *Nazwisko*, *Imię*, *Adres*, *Telefon*, *PESEL*; *Zamówienia*, z polami *Nazwisko klienta*, *Tytuł książki*, *Liczba egzemplarzy*; oraz *Książki*, z polami *Tytuł książki* i *Cena* (rysunek 1.33). Kluczem podstawowym w tabeli *Klient* są pola *Nazwisko* i *Imię*, w tabeli *Zamówienia* pola *Nazwisko klienta* i *Tytuł książki*, a w tabeli *Książki* pole *Tytuł książki*. Zostało zlikwidowane powtarzanie danych w tabeli *Realizacja zamówień* i wszystkie tabele są w II PN.

Nazwisko klie	Imię	Adres	Telefon	PESEL
Nowak	Marek	Toruń	(56) 6589234	79120307431
Kowalski	Adam	Warszawa	(22) 3451234	80122401871
Górecki	Grzegorz	Poznań	(45) 2367897	82061203983
Zan	Marcin	Gdańsk	(33) 8373635	82020201875
Bagińska	Anna	Warszawa	(22) 7336252	91110402837
Pol	Aleksander	Szczecin	(23) 6517830	70073003228

Nazwisko klient	Tytuł książki	Liczba egz.
Nowak	Dziady	2
Kowalski	Balladyna	1
Kowalski	Tango	2
Górecki	Pan Tadeusz	1

Tytuł książki	Cena
Dziady	20,00 zł
Balladyna	15,00 zł
Tango	18,00 zł
Pan Tadeusz	21,00 zł

Rysunek 1.33. Podział tabeli *Realizacja zamówień* na tabele spełniające warunek II PN

Aby wyświetlić zestawienie dotyczące klienta i kupionych przez niego książek, należy zdefiniować połączenie między tabelami.

Trzecia postać normalna

DEFINICJA

Tabela jest w trzeciej postaci normalnej (III PN), jeżeli jest w pierwszej i w drugiej postaci normalnej oraz każde z pól niewchodzących w skład klucza podstawowego niesie informację bezpośrednio o kluczu i nie odnosi się do żadnego innego pola.

Załóżmy, że w projektowanej bazie danych dla księgarni została zaprojektowana tabela *Faktura* z polami: *Nazwisko klienta*, *Imię*, *Adres*, *PESEL*, *Numer faktury*, *Sposób płatności* i *Data wystawienia faktury*. Załóżmy również, że w pola tabeli będą wpisywane tylko

wartości elementarne, czyli tabela jest w I PN. Klucz podstawowy to pole *Numer faktury*. Wszystkie pola niewchodzące w skład klucza zależą od całego klucza, czyli tabela jest w II PN. Sprawdźmy, czy tabela jest w III PN. Pola *Sposób płatności* i *Data wystawienia faktury* odnoszą się do faktury, czyli zawierają informacje o kluczu. Natomiast pola *Adres* i *PESEL* zawierają informacje na temat klienta, a nie faktury (rysunek 1.34), czyli nie niosą informacji bezpośrednio o kluczu. Tabela nie jest w III PN.

Klucz podstawowy

Nazwisko klie	Imię	Adres	PESEL	Numer faktury	Sposób płatn	Data wystaw
Nowak	Marek	Toruń	79120307431	1	gotówka	2013-04-06
Kowalski	Adam	Warszawa	80122401871	2	przelew	2012-07-29
Kowalski	Adam	Warszawa	80122401871	4	gotówka	2012-01-17
Bagińska	Anna	Warszawa	91110402837	5	gotówka	2012-08-29
Pol	Aleksander	Szczecin	70073003228	6	gotówka	2012-03-02
Pol	Aleksander	Szczecin	70073003228	3	przelew	2012-12-10

Odnoszą się do pól *Nazwisko* i *Imię* Odnoszą się do pola *Numer faktury*

Rysunek 1.34. Tabela nie jest w III PN, ponieważ pola *Adres* i *PESEL* nie niosą informacji o kluczu

Normalizacja, podobnie jak w przypadku II PN, polega na podzieleniu tabeli na takie tabele, które spełniają warunek III PN.

Tabelę *Faktura* należy podzielić na dwie tabele: *Faktura* (z polami *Numer faktury*, *Sposób płatności* i *Data wystawienia faktury*) oraz *Klient* (z polami *Nazwisko klienta*, *Imię*, *Adres*, *PESEL*) (rysunek 1.35). Zostało zlikwidowane powtarzanie danych o kliencie w tabeli *Faktura*. Dane o kliencie będą zapisane tylko raz, w tabeli *Klient*.

Nazwisko klie	Imię	Adres	PESEL
Nowak	Marek	Toruń	79120307431
Kowalski	Adam	Warszawa	80122401871
Górecki	Grzegorz	Poznań	82061203983
Zan	Marcin	Gdańsk	82020201875
Bagińska	Anna	Warszawa	91110402837
Pol	Aleksander	Szczecin	70073003228

Numer faktury	Sposób płatn	Data wystawie:	Nazwisko klienta
1	gotówka	2013-04-06	Nowak
2	przelew	2012-07-29	Kowalski
3	przelew	2012-12-10	Pol
4	gotówka	2012-01-17	Kowalski
5	gotówka	2012-08-29	Bagińska
6	gotówka	2012-03-02	Pol

Rysunek 1.35. Podział tabeli *Faktura* na tabele spełniające warunek III PN

Przykład 1.2

Przestrzegając reguł tworzenia tabel, po sprawdzeniu za pomocą normalizacji, czy tabele mają prawidłową strukturę, baza danych dla księgarni mogłaby składać się z następujących tabel:

Klient → *Id_klienta*, *Nazwisko*, *Imię*, *Kod pocztowy*, *Miejscowość*, *Ulica*, *Nr domu*, *PESEL*, *Telefon*, *Adres e-mail*.

Zamówienia → *Id_zamówienia*, *Id_klienta*, *Data złożenia zamówienia*, *Data wysłania*, *Koszt wysyłki*.

Rejestr → *Id_zamówienia*, *Id_książki*, *Liczba egzemplarzy*.

Książka → *Id_książki*, *Tytuł*, *Id_autora*, *Cena książki*, *Rok wydania*, *Wydawnictwo*, *Rodzaj literatury*, *Miejsce wydania*, *Język książki*, *Opis*, *Zdjęcie okładki*.

Autor → *Id_autora*, *Nazwisko*, *Imię*, *Narodowość*, *Okres tworzenia*, *Rodzaj twórczości*, *Język tworzenia*, *Osiągnięcia*.

1.4.8. Prawidłowy projekt bazy danych

Prawidłowy projekt bazy danych jest bardzo istotny dla efektywnej pracy, dlatego warto poświęcić trochę czasu, aby opanować zasady projektowania bazy.

Dobry projekt nie powinien zawierać powtarzających się danych. Aby osiągnąć ten cel, musimy podzielić dane na wiele tabel. Następnie powinniśmy zdefiniować połączenia między tabelami, aby można było tworzyć zestawienia danych pochodzących z różnych tabel. Na przykład w bazie danych *Księgarnia* podzieliliśmy dane na oddzielne zbiory (*Klient*, *Zamówienia*, *Rejestr*, *Książka*, *Autor*, *Faktura*), następnie zdefiniowaliśmy połączenia między tabelami, aby utworzyć zestawienie dotyczące realizacji zamówienia (*Nazwisko*, *Imię*, *Tytuł książki*, *Nazwisko autora*, *Cena książki*, *Liczba egzemplarzy*).

Proces projektowania bazy danych składa się z następujących kroków:

- **Określenie celu, jakiemu ma służyć baza danych.** Baza danych może na przykład służyć do gromadzenia informacji na temat sprzedaży książek, do wystawiania faktur dotyczących sprzedaży, do modyfikowania na bieżąco tych danych, do przetwarzania zgromadzonych danych.
- **Określenie zakresu potrzebnych informacji.** Należy określić, jakie informacje będą przechowywane w bazie, na przykład: nazwisko i imię klienta oraz jego dane osobowe, tytuły książek, informacje o autorach, informacje na temat realizacji zamówień.
- **Podzielenie informacji na tabele.** Zebrane informacje należy podzielić według tematów i dla każdego przewidzieć oddzielną tabelę, na przykład *Klient*, *Książki*.
- **Podzielenie elementów informacji na kolumny.** Trzeba zdecydować, jakie informacje mają być przechowywane w poszczególnych tabelach. Każdy element informacji zostanie przypisany do kolumny, na przykład tabela *Klient* będzie zawierała kolumny *Nazwisko klienta* i *Adres*.
- **Wybranie kluczy podstawowych.** Należy wybrać klucz podstawowy dla każdej tabeli, na przykład w tabeli *Klient* może to być identyfikator przypisany do każdego klienta lub PESEL.
- **Zastosowanie reguł normalizacji.** Za pomocą reguł normalizacji można sprawdzić, czy tabele mają prawidłową strukturę.

- **Poprawienie projektu.** Po sprawdzeniu, jeżeli to konieczne, trzeba skorygować projekt bazy.
- **Utworzenie relacji pomiędzy tabelami.** Należy przejrzeć projekt i zdecydować, jakie relacje powinny znaleźć się w bazie.

Po zaprojektowaniu bazy danych zgodnie z podanymi regułami można przystąpić do jej tworzenia, korzystając z aplikacji przeznaczonych do obsługi relacyjnych baz danych.

Przykład 1.3

Podczas analizowania przeznaczenia bazy danych tworzymy jej strukturę. Jeżeli w bazie danych dla księgarni zmienimy jej przeznaczenie, może okazać się, że tabela *Autor* nie jest potrzebna. Natomiast konieczne jest sporządzanie dla każdej sprzedaży faktury. Wtedy niezbędna będzie tabela do przechowywania informacji, które powinny znaleźć się na fakturze.

Baza danych mogłaby składać się z następujących tabel:

Klient → *Id_klienta*, *Nazwisko*, *Imię*, *Kod pocztowy*, *Miejscowość*, *Ulica*, *Nr domu*, *PESEL*, *Telefon*, *Adres e-mail*.

Zamówienia → *Id_zamówienia*, *Id_klienta*, *Data złożenia zamówienia*, *Data wystania*, *Koszt wysyłki*, *Numer faktury*.

Rejestr → *Id_zamówienia*, *Id_książki*, *Liczba egzemplarzy*.

Książka → *Id_książki*, *Tytuł*, *Nazwisko i imię autora*, *Cena książki*, *Wydawnictwo*, *Rodzaj literatury*, *Miejsce wydania*, *Język książki*, *Opis*.

Faktura → *Numer faktury*, *Sposób płatności*, *Data wystawienia faktury*.

Ćwiczenie 1.1

Firma Globtroter wynajmuje klientom autokary do przewozu osób. Posiada flotę dobrze wyposażonych autokarów oraz zatrudnia grupę kierowców z najwyższymi umiejętnościami. Utwórz projekt graficzny bazy danych, która będzie służyła do rejestracji usług świadczonych przez tę firmę. W bazie danych powinna znaleźć się informacja o posiadanych autokarach, pracujących w firmie kierowcach oraz klientach korzystających z usług firmy. Każda usługa zamówienia może dotyczyć jednego lub kilku autokarów i powinna zostać zarejestrowana w bazie z informacją, dla kogo realizowane jest zamówienie, które autokary zostaną wysłane na trasę i którzy kierowcy będą je obsługiwali.

Rozwiązanie

Określenie potrzebnych informacji:

Przechowywane informacje o autokarach: model, liczba miejsc, rok produkcji, pojemność silnika, spalanie, stawka za 1 km.

Przechowywane informacje o kierowcach: nazwisko, imię, PESEL, adres, telefon, kategoria prawa jazdy, data zatrudnienia, doświadczenie.

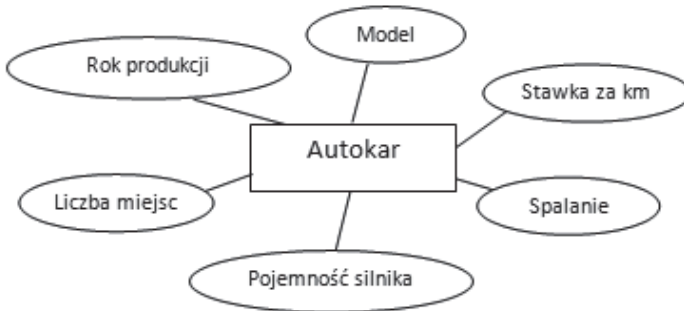
Przechowywane informacje o klientach: nazwisko, imię, PESEL, adres, telefon.

Przechowywane informacje o realizowanych usługach: cel, liczba km, data realizacji, opłata za wynajęcie.

Przechowywane informacje o fakturach: nr faktury, data wystawienia, sposób płatności.

Tworzenie diagramu ERD:

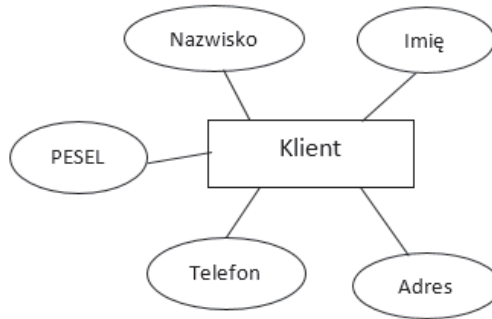
Tworzone zbiory encji zostały pokazane na rysunkach 1.36–1.40.



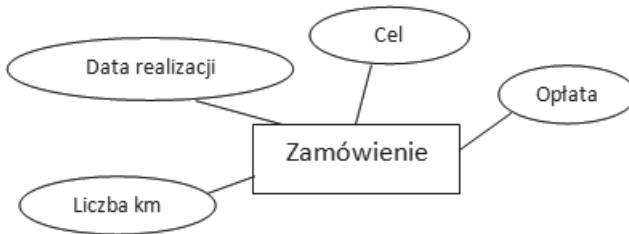
Rysunek 1.36. Zbiór encji Autokar



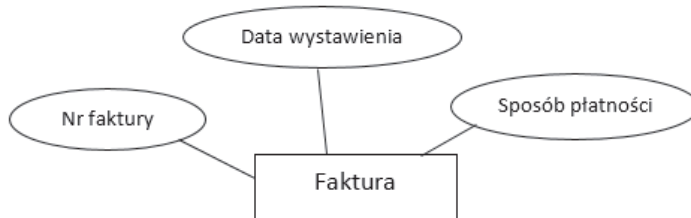
Rysunek 1.37. Zbiór encji Kierowca



Rysunek 1.38. Zbiór encji Klient



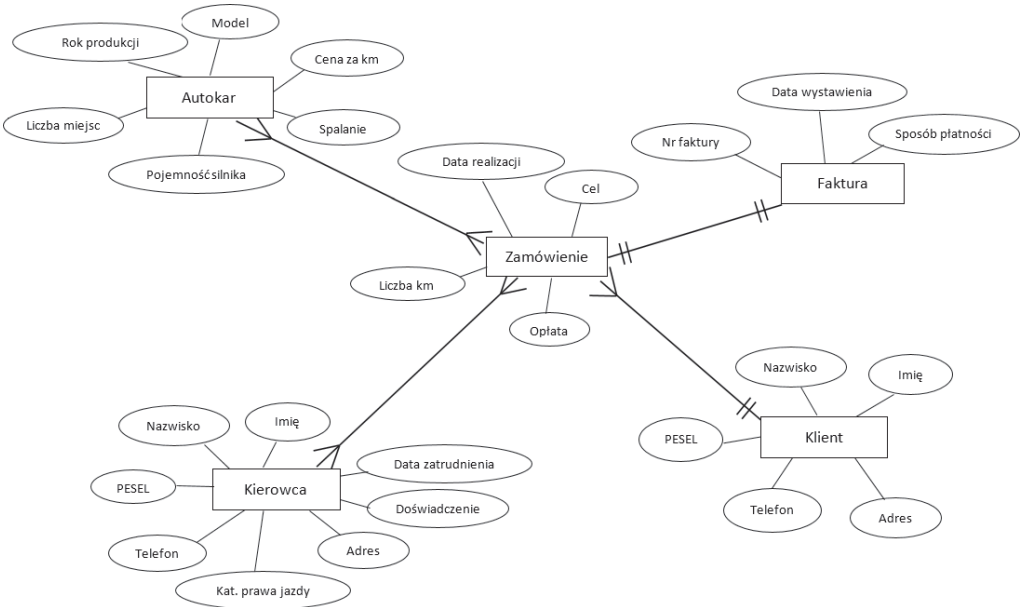
Rysunek 1.39. Zbiór encji Zamówienie



Rysunek 1.40. Zbiór encji Faktura

Definiowanie związków między zbiorami encji:

Tworzone związki zostały pokazane na rysunku 1.41.



Rysunek 1.41. Związki między zbiorami encji bazy danych dla firmy Globtroter

Ćwiczenie 1.2

Na podstawie utworzonego w ćwiczeniu 1.1 projektu graficznego bazy danych dla firmy Globtroter zaprojektuj w modelu relacyjnym tabele do przechowywania danych oraz tabele opisujące związki. Umieść w nich wszystkie atrybuty zdefiniowanych encji oraz atrybuty opisujące związki zachodzące między encjami.

Rozwiązanie

Przykładowy zestaw tabel dla bazy danych firmy Globtroter:

Klient → *Id_klienta*, *Nazwisko*, *Imię*, *Kod pocztowy*, *Miejscowość*, *Ulica*, *Nr domu*, *PESEL*, *Telefon*.

Zamówienia → *Id_zamówienia*, *Id_klienta*, *Data realizacji*, *Cel*, *Oplata*, *Liczba km*, *Nr faktury*.

Kierowca → *Id_kierowcy*, *Nazwisko*, *Imię*, *Kod pocztowy*, *Miejscowość*, *Ulica*, *Nr domu*, *PESEL*, *Telefon*, *Data zatrudnienia*, *Doświadczenie*, *Kat. prawa jazdy*.

Autokar → *Id_autokaru*, *Model*, *Rok produkcji*, *Liczba miejsc*, *Pojemność silnika*, *Spalanie*, *Stawka za km*.

Faktura → *Nr faktury*, *Sposób płatności*, *Data wystawienia*.

Zamówienie-Kierowca → *Id_zamówienia*, *Id_kierowcy*.

Zamówienie-Autokar → *Id_zamówienia*, *Id_autokaru*.

Skorowidz

A

abstrakcja danych, 48
Access, 51
definiowanie wyrażeń, 68
drukowanie raportu, 116
formanty, 88
formanty listy, 106
formularze, 52, 87
indeksowanie, 68
interfejs użytkownika, 52
kreator przycisków poleceń, 94
kwerendy, 52
makra, 52, 95
moduły, 52, 116
obiekty programu, 52
okno
 konstruktora wyrażeń, 72
 nawigacji, 53
 projektu tabeli, 54
 Relacje, 83
 Właściwości sprzężenia, 84
projekt kwerendy, 76
raporty, 52, 112
reguły integralności, 62
tabele, 52
typy danych, 55
właściwości pól tabeli, 58
administrator baz danych, 233
administrowanie serwerami, 233
akcje makr, 97
aktualizowanie danych, 185
algebra relacji, 13
aliasy, 179
ANSI, 139
architektura
 3-warstwowa, 121
 klient-serwer, 120
 systemu baz danych, 120
atrybut, 19, 22
 AUTO_INCREMENT, 155
 CHECK, 159
 DEFAULT, 157
 IDENTITY, 155
 NOT NULL, 154
 UNIQUE, 158, 160
atrybuty kolumn, 153
autoryzacja, 235

B

bazy danych
 architektura 3-warstwowa, 121
 architektura klient-serwer, 120
 formularz sterujący, 112
 hierarchia obiektów, 144
 indeksy, 267
 konceptualne projektowanie, 21
 lokalne, 51
 modele, 8
 naprawa, 261
 normalizacja, 267
 operacje wykonywane, 269
 optymalizacja, 267
 pełna kopia, 247, 262
 planowanie, 19
 prawidłowy projekt, 38
 procedury składowane, 270
 projektowanie, 7, 18
 przyrostowa kopia, 247, 264
 przywracanie, 249
 reguły integralności, 62
 relacyjne, 7, 11, 15, 46
 schemat, 29
 sprawdzanie spójności, 248, 261
 struktura, 270
 systemowe, 237
 systemy zarządzania, 119
 tworzenie, 238
 typy danych, 43
 zarządzanie, 237
 zmiana parametrów, 238
bezpieczeństwo danych, 47
blokady
 tryby, 205
 współdzielone, 205
 wyłączne, 205
 zakresy, 206
blokowanie danych, 205
brudne odczyty, 206

C

CASE, Computer Aided Software Engineering, 25
cechy relacyjnej bazy danych, 46

D

DBMS, Database Management System, 121
 DCL, Data Control Language, 140, 141
 DDL, Data Definition Language, 140, 141
 definiowanie
 klucza obcego, 182
 klucza podstawowego, 55
 makr, 96
 relacji, 11
 wyrażeń, 68
 diagram
 ERD, 21, 24
 związków encji, 23
 DML, Data Manipulation Language, 140, 141, 160
 dodanie kolumny, 152
 DQL, Data Querying Language, 141
 druga postać normalna, 35
 drukowanie raportu, 116
 dziedziczenie uprawnień, 245
 dziedzina, 19

E

eksportowanie
 danych, 250, 264
 projektu, 29
 elementy wyrażenia, 69, 73
 encja, 19, 21
 ERD, Entity Relationship Diagram, 21
 etykiety, 89

F

formanty
 etykiety, 89
 listy, 106
 pole kombi, 106
 pole listy, 106
 pole tekstowe, 89
 właściwości, 92
 formularze, 52, 87
 ciągłe, 102
 do przeglądania danych, 99
 do wprowadzania danych, 100
 funkcje, 99
 obiektów graficznych, 92
 pojedyncze, 102
 przyciski poleceń, 93
 sterujące, 111
 właściwości, 92
 z listą wartości, 109
 z podformularzem, 103
 z polami kombi, 108
 funkcje, 70
 agregujące, 172

formularza, 99
 składowane, 218
 wbudowane, 220

G

graficzna reprezentacja
 encji, 21
 opcjonalności związku, 23
 związków, 22
 grupowanie danych, 172

H

hierarchia obiektów, 144
 hiperłącze, 57

I

identyfikatory, 69
 iloczyn kartezjański, 14, 178
 importowanie danych, 250, 266
 indeksowanie, 68
 indeksy, 198, 267
 mieszane, 255
 instalowanie serwera
 MySQL, 123, 125
 SQL Server 2017, 130
 instrukcja
 COMMIT, 269
 DELETE, 163
 DENY, 245
 DQL, 141
 EXCEPT, 188
 GRANT, 244, 257
 INSERT, 160
 INTERSECT, 188
 REVOKE, 244, 258
 SELECT, 164
 klauzule, 165
 wyrażenia, 166
 UNION, 187
 UPDATE, 162
 instrukcje
 DCL, 140, 141
 DDL, 140, 141
 DML, 140, 141
 języka SQL, 141
 TCL, 141
 warunkowe, 213
 integralność, 46, 62
 danych, 13, 17, 49
 encji, 13
 referencyjna, 13
 interfejs użytkownika, 52
 izolowanie transakcji, 206, 207

J

język
definiowania danych, DDL, 145
SQL, 73

K

kaskadowe
usuwanie danych, 185
aktualizowanie danych, 186

klauzula
CONSTRAINT, 184
DISTINCT, 165
FROM
podzapytania, 194
GROUP BY, 174
HAVING, 174
ORDER BY, 167
REFERENCES, 184
TOP, 171
WHERE, 168
podzapytania, 190

klauzule dla instrukcji SELECT, 165

klient-serwer, 120

klucz, 13
obcy, 60, 182
podstawowy, 16, 55, 60, 153
sztuczny, 17

kod SQL wygenerowany automatycznie, 30

konceptualne projektowanie, 21

konstrukcja zapytania, 165

konstruktor
makr, 96
wyrażeń, 72

kontrola współbieżności, 205

kopia bezpieczeństwa, 246
pełna, 247, 262
przyrostowa, 247, 264

kreator, 57
formularzy, 105
pól kombi, 107
przycisków poleceń, 94
raportu, 114

kwerendy, 52, 73
krzyżowe, 81
podsumowujące, 77
wybierające, 74
wybór typu sprzężenia, 83
z parametrem, 80
z polem wyliczeniowym, 79

L

lista
uprawnień, 258
wartości pola kombi, 109

listy, 107

lokalne bazy danych, 51

Ł

łączenie
tabel, 176
wyników zapytań, 187

M

makra, 52, 95
akcje, 97
kreator formularzy, 105
wiązanie ze zdarzeniami, 97

manipulowanie danymi, DML, 160

maska wprowadzania, 59, 65

model baz danych, 8
hierarchiczny, 9
obiektowy, 10
relacyjny, 10, 11
sieciowy, 9

modele konceptualne, 21

moduły, 52, 116

modyfikowanie
kolumny, 152
widoku, 197

MS SQL Server, 129

MySQL, 253
konfigurowanie serwera, 253
kopie bezpieczeństwa, 261
optymalizacja wydajności, 266
pełne kopie, 262
poprawa wydajności serwera, 271
prawa dostępu, 256
przyrostowe kopie, 264
tryby uwierzytelnienia, 253
typy tabel, 254
zarządzanie bazami danych, 254

N

nadawanie
praw, 257
uprawnień, 244

naprawa bazy danych, 261

narzędzia CASE, 25

narzędzie, *Patrz* program

niepowtarzalne odczyty, 206

niezależność danych, 48

normalizacja, 267
tabel, 34

O

- obiekt OLE, 57
- obraz, 93
- odbieranie
 - praw, 258
 - uprawnień, 244
- odczyty widma, 207
- odzyskiwanie danych, 264
- okno
 - edytowania połączenia, 28
 - edytowania tabeli, 26
 - kolejności tabulacji, 101
 - konstruktora wyrażeń, 72
 - narzędzia phpMyAdmin, 128
 - projektu raportu, 113
 - projektu tabeli, 54
 - właściwości sprzężenia, 84
- operator, 70
 - ALL, 194
 - ANY, 193
 - EXISTS, 192
 - SOME, 193
- operatory podzapytań, 192
- optymalizacja
 - bazy danych, 267
 - wydajności serwera, 270
 - wydajności SZBD, 266
 - zapytań, 269
- połączenia, 17
 - krzyżowe, 178
 - rozszerzające, 84
 - wewnętrzne, 176
 - wielokrotne, 178
 - zawężające, 84
 - zewnętrzne, 176, 177
- poziomy izolowania transakcji, 207
- prawa dostępu do serwera, 240, 256
- procedury
 - składowane, 215, 270
 - wbudowane, 218
- program
 - DBDesigner4, 27
 - MS Access, 51
 - phpMyAdmin, 127
 - SQL Server Management Studio, 134
- projekcja, 14
- projektowanie
 - bazy danych, 18
 - formularzy, 87
 - listy, 107
 - raportu, 114
 - tabeli, 54
- przyciski
 - operatorów, 73
 - poleceń, 93
- przypisywanie do ról, 242
- przywracanie bazy danych, 249
- punkty przywracania, 204

P

- pakiet
 - MySQL, 125
 - XAMPP, 127
- pamięć operacyjna, 270
- perspektywy, *Patrz* widoki
- phpMyAdmin, 127
- pierwsza postać normalna, 34
- podzapytania, 189
 - klauzuli FROM, 194
 - klauzuli WHERE, 190
 - skorelowane, 196
 - w instrukcjach modyfikujących dane, 195
- pole
 - kombi, 106, 108
 - wprowadzanie danych, 108
 - wyszukiwanie rekordu, 110
 - listy, 106
 - tekstowe, 89
 - wyrażenia, 73
- polecenie
 - BULK INSERT, 251
 - mysqldump, 262
 - REPAIR TABLE, 261

Q

- QBE, Query By Example, 73

R

- raporty, 52, 112
- Read Committed, 208
- Read Uncommitted, 207
- reguły
 - integralności, 62
 - poprawności, 59, 67
- relacja, 11, *Patrz* także związek
 - „jeden do jednego”, 18, 31
 - „jeden do wielu”, 27, 61
 - „wiele do jednego”, 18, 32
 - „wiele do wielu”, 28, 33
- relacje między tabelami, 60, 83
- relacyjna baza danych, 7, 11, 15
- Repeatable Read, 209
- role, 240, 258
 - bazodanowe, 241
 - serwerowe, 241

S

samopłączenie, 84
 schemat, 149

- bazy danych, 29
- relacji, 12

 selekcja, 13
 serializable, 210
 serwery bazodanowe, 122, 233

- MySQL, 123, 125, 253, 271
- SQL Server, 270

 składnia języka SQL, 140
 skrypty tworzące tabele, 228, 230
 słowo kluczowe INNER JOIN, 176
 sortowanie, 167
 spójność bazy danych, 248, 261
 sprawdzanie poprawności danych, 64
 sprzężenia, 83

- rozszerzające, 85, 86

 SQL, Structured Query Language, 73, 139

- dialekty języka, 140
- składnia języka, 140
- standardy języka, 139
- terminatory, 140
- typy danych, 142

 SQL Server

- Management Studio, 134
- optymalizacja wydajności, 270

 stałe, 70
 standard

- ANSI, 139
- SQL99, 139

 struktura

- bazy danych, 270
- tabeli, 16, 129

 systemowe bazy danych, 237
 systemy zarządzania bazą danych, 8, 119
 SZBD, Database Management Systems, 8, 119

Ś

środowisko SQL Server, 234

T

tabele, 15, 52, 53

- klucz podstawowy, 16
- łączenie, 176
- pochodne, 194
- tworzenie, 147
- usuwanie, 147
- w MySQL, 254
- właściwości pól, 58
- zmiana struktury, 152

 TCL, Transaction Control Language, 141

terminatory SQL, 140
 transakcje, 199, 269

- Autocommit, 201
- Explicit, 200
- Implicit, 202
- izolowanie, 206
- poziomy izolowania, 207
- zagnieżdżanie, 203

 transformacja

- encji do tabeli, 31
- modelu konceptualnego, 31

 trwałość danych, 46
 tryb

- niezatwierdzonego odczytu, 207
- odczytu zatwierdzonego, 208
- powtarzalnego odczytu, 209
- szeregowania, 210

 tryby uwierzytelniania, 234, 253
 trzecia postać normalna, 36
 tworzenie

- bazy danych, 238, 254
- kopii bezpieczeństwa, 247, 262
- modelu konceptualnego, 21
- ról, 242
- tabel, 147
- widoków, 197

 typy

- danych, 43, 55, 142
 - binarne, 45
 - daty i czasu, 45
 - liczbowe, 44
 - specjalne, 46
 - tekstowe, 43
 - walutowe, 45
- połączeń, 18

U

uprawnienia, 223, 243, 258

- dziedziczenie, 245
- nadawanie, 244
- odbieranie, 244

 usuwanie

- kolumn, 152
- ról, 242
- tabel, 147
- widoku, 197

 utrata aktualizacji, 206
 uwierzytelnianie, 234, 235

W

wartość

- domyślna, 59, 64
- NULL, 14, 144

widoki, 196, 232
 modyfikowanie, 197
 tworzenie, 197
 usuwanie, 197

więzy
 integralności, 182
 ogólne, 13

właściciel obiektu, 246

właściwości
 formantów, 92
 formularza, 92
 pól tabeli, 58
 transakcji, 199
 zdarzeń, 95

wprowadzanie danych, 57, 108

współbieżność, 204

współdzielenie danych, 48

wybór
 atrybutów, 14
 krotek, 13
 typu sprzężenia, 83
 wierszy, 168, 171

wydajność SZBD, 266

wyrażenia, 68
 CASE, 214
 funkcje, 70
 identyfikatory, 69
 konstruktor, 72
 operatory, 70
 stałe, 70
 tablicowe, CTE, 214
 w instrukcji SELECT, 166
 wartości/literały, 70

wyszukiwanie rekordu, 110

wyzwalacze, triggers, 220
 DDL, 222
 DML, 221
 logowania, 223

X

XAMPP, 127

Z

zadania administratora, 233

zagnieżdżanie transakcji, 203

zakleszczenia, 206

zakresy blokad, 206

załącznik, 57

zapytania, *Patrz* kwerendy, 52

zarządzanie bazami danych, 8, 119, 237

zbiór encji, 40, 41

zdarzenia
 właściwości, 95

złączenia, *Patrz* połączenia, 177

złączenie dwóch relacji, 14

zmiana struktury tabeli, 152

zmienne, 211
 systemowe, 212

związek
 „jeden do jednego”, 18, 31
 „jeden do wielu”, 27, 61
 „wiele do jednego”, 18, 32
 „wiele do wielu”, 28, 33
 opcjonalny, 23
 wymagany, 23

związki między zbiorami, 42

PROGRAM PARTNERSKI

— GRUPY HELION —



1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA
Helion 



Kwalifikacja EE.09

Programowanie, tworzenie i administrowanie stronami internetowymi i bazami danych

Podręcznik do nauki zawodu **technik informatyk**

Technik informatyk to nie tylko tytuł uzyskany po ukończeniu szkoły średniej, ale i zawód będący przepustką do kariery. Dzięki solidnej podstawie teoretycznej i bieżącym uaktualnieniom podręcznika uczniowie — już jako absolwenci — łatwo nawiążą współpracę z firmami informatycznymi, także tymi, które specjalizują się w tworzeniu baz danych i administrowaniu nimi. Autorka książki opracowała pozycję na wysokim poziomie merytorycznym, z licznymi przykładami i zadaniami umożliwiającymi praktyczne zastosowanie podanych informacji.

Budowa podręcznika i poszczególnych rozdziałów pozwala na realizację treści programowych w sposób wybrany przez nauczyciela oraz na samodzielną pracę ucznia. Książka porusza pięć głównych zagadnień, ułożonych w logicznie rozplanowanych rozdziałach. Na początek uczeń przyswoi zasady projektowania relacyjnych baz danych, opanuje dotyczące ich słownictwo i dowie się, w jakich sytuacjach są one stosowane. Następnie pozna program MS Access, służący do tworzenia lokalnych baz danych i zarządzania nimi. W rozdziale trzecim zgłębi tematykę architektury systemów baz danych. Nauczy się instalować i konfigurować systemy bazodanowe, również te używane do zarządzania bazami danych. Kolejną część podręcznika poświęcono strukturalnemu językowi zapytań SQL i jego roli w pracy z bazami danych. Na koniec omówiono temat administrowania bazami danych, zarządzania uprawnieniami użytkowników oraz kontroli ustawień dostępu, a także tworzenia kopii zapasowych i odzyskiwania utraconych danych.

Technik Informatyk to charakteryzujący się wysoką jakością kompletny zestaw edukacyjny, przygotowany przez dysponującego ogromnym doświadczeniem lidera na rynku książek informatycznych — wydawnictwo Helion.

W skład zestawu do nauki w ramach kwalifikacji EE.09 wchodzi także:

- *Kwalifikacja EE.09. Programowanie, tworzenie i administrowanie stronami internetowymi i bazami danych. Część 1. Tworzenie stron internetowych. Podręcznik do nauki zawodu technik informatyk*
- *Kwalifikacja EE.09. Programowanie, tworzenie i administrowanie stronami internetowymi i bazami danych. Część 2. Programowanie aplikacji. Podręcznik do nauki zawodu technik informatyk*
- *Kwalifikacja EE.09. Programowanie, tworzenie i administrowanie stronami internetowymi i bazami danych. Część 4. Tworzenie aplikacji internetowych. Podręcznik do nauki zawodu technik informatyk*

Podręczniki oraz inne pomoce naukowe należące do tej serii zostały opracowane z myślą o wykształceniu kompetentnych techników, którzy bez trudu poradzą sobie z wyzwaniami, jakie stawia przed nimi współczesna informatyka. Wiedza zawarta w serii pomoże zdać egzamin zawodowy i uzyskać umiejętności praktyczne, przydatne w przyszłej pracy.

Helion

helion.pl

HELION SA
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
helion@helion.pl

Sprawdź nasze szkolenia!



AKADEMIA IT & BUSINESS

WWW.SZKOLENIA.HELION.PL

KOD KORZYŚCI
Stęgnij po więcej!



ISBN 978-83-283-4863-9

