

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.

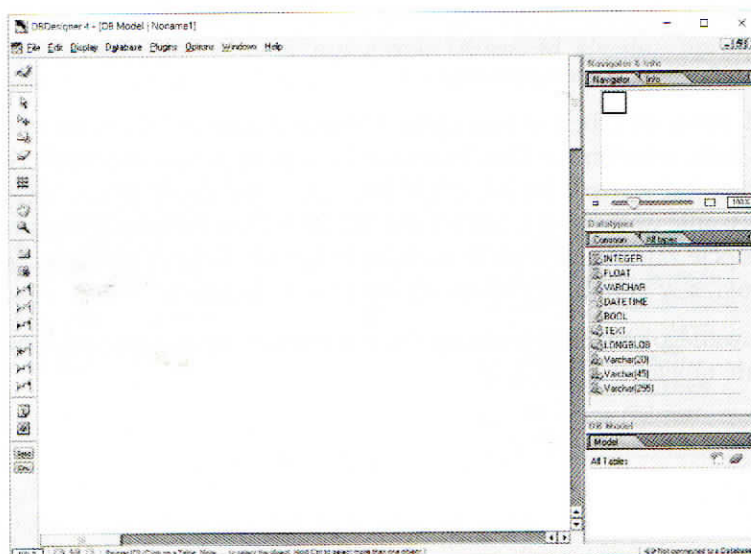
Narzędzia te pozwalają tworzyć modele graficzne odpowiadające konstrukcjom programistycznym. Wykorzystywane są tutaj edytory notacji graficznych, które dają możliwości tworzenia diagramów i powiązań między poszczególnymi elementami. Bardziej zaawansowane edytory umożliwiają przetwarzanie informacji i 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).

Table Editor

Table Name: Klient Table Prefix: Default (no prefix) Table Type: MYISAM (Standard) Weak entity: ☐ is rnm Table

Column Name	Data Type	NN	AI	Flags	Default Value	Comments
id_klienta	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL		
Nazwisko	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY		
Imię	VARCHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY		
Adres	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY		
Telefon	VARCHAR(9)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY		
E-mail	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY		

Indices

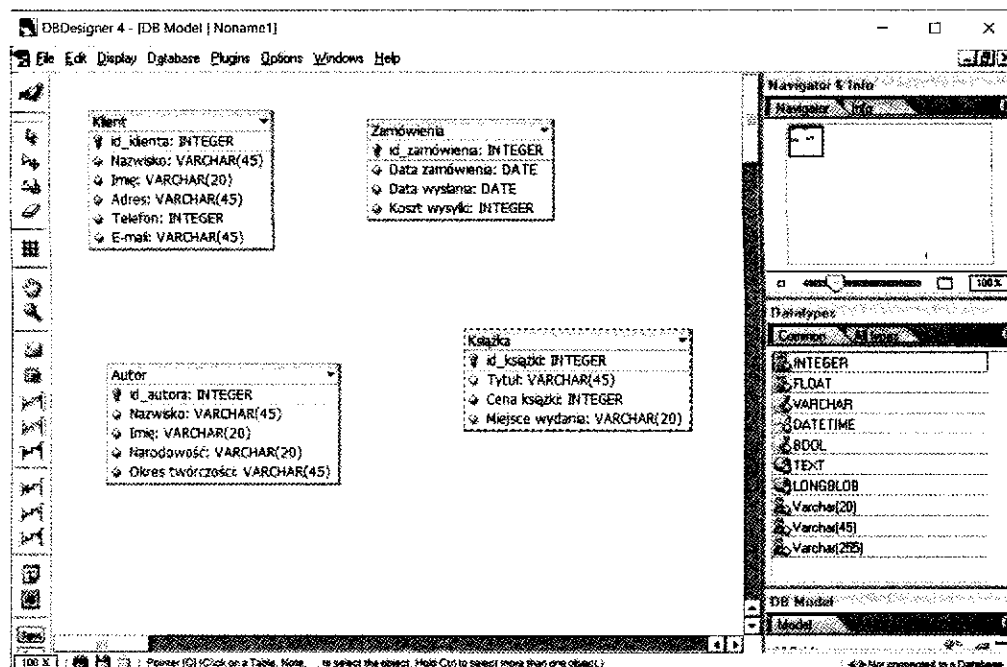
Table Options: ☒ Advanced ☐ Standard Inserts ☐ Comments

Indexname: PRIMARY Index Type: PRIMARY Columns (use Drag'n'Drop to add Columns): id_klienta

Rysunek 1.17. Okno edytowania tabeli

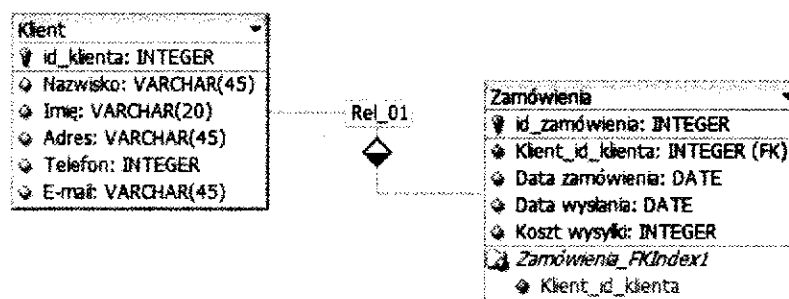
W oknie edytowania tabeli w kolumnie *Column Name* należy wprowadzić nazwę tworzonego pola, w kolumnie *Data Type* określić typ danych, w kolumnie *NN* określić, czy dozwolona jest wartość *NULL* (*NOT NULL*), w kolumnie *AI* zaznaczyć automatyczne zwiększanie wartości o 1 (*AUTO INCREMENT*), w kolumnie *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).

Relation Editor

Relation Name: Klient_Zamowienia Relation Kind: 1:n (Nonidentifying) Visibility: Visible

Tables:

Source: Klient Destination: Zamowienia

Foreign Keys:

Source Column	Dest. Name	Comment
id_klienta	Klient_id_klienta	

Reference Definition:

☒ Create Reference Definition

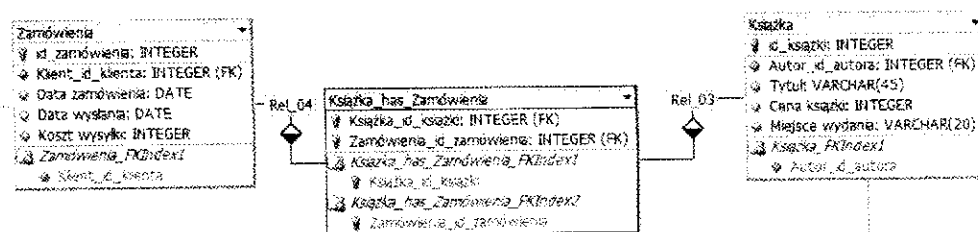
On Delete: NO ACTION

On Update: NO ACTION

Rysunek 1.20. Okno edytowania połączenia

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

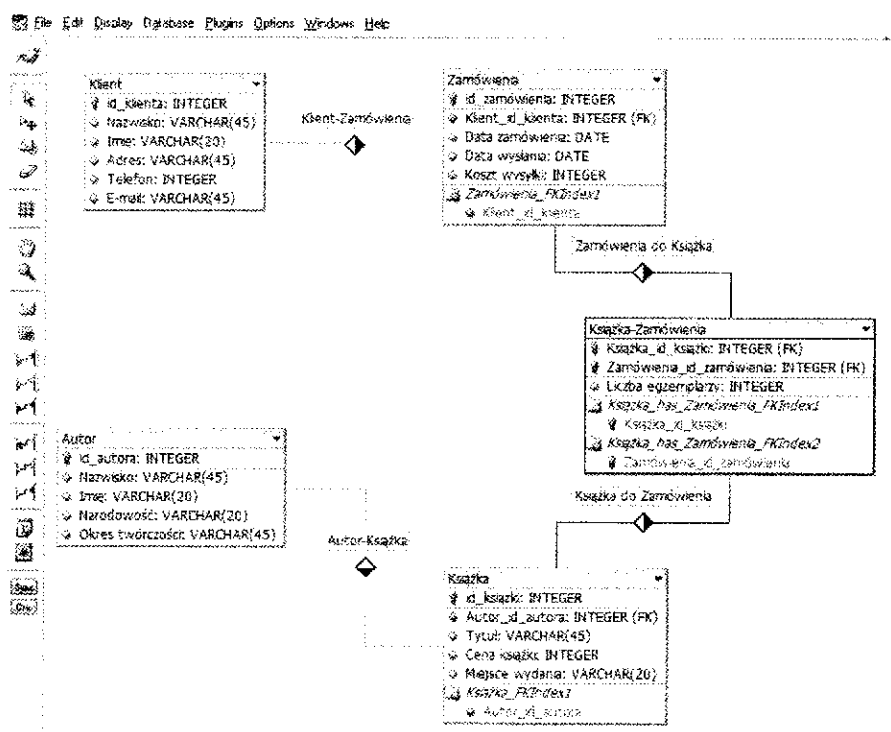
Ponieważ w jednym zamówieniu może znaleźć się kilka książek, a jedna książka może zostać wskazana w wielu zamówieniach, między tabelami *Zamówienia* i *Książki* występuje połączenie „wiele do wielu”. Aby dodać taki typ połączenia, należy wybrać ikonę *New n:m*. Po wybraniu tej ikony klikamy na przykład 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.

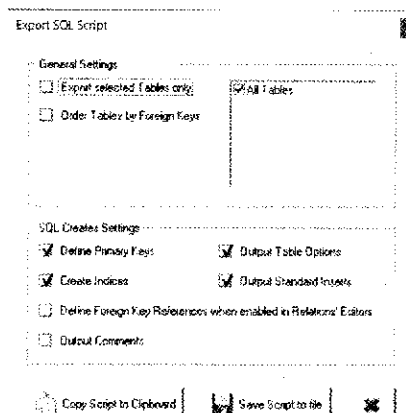
Po utworzeniu połączeń i dodaniu potrzebnych atrybutów uzyskamy efekt podobny do pokazanego na rysunku 1.22.



Rysunek 1.22. Schemat bazy danych uzyskany w programie DBDesigner4

Utworzony projekt należy zapisać w pliku, wybierając z menu *File/Save*. Można również zaimportowaną 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(30) 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(30) 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 bazodanowych w celu wygenerowania gotowej bazy danych.