

## 6.4. Zaawansowane możliwości programów obsługi baz danych

### 6.4.1. Import i eksport danych

Programy bazy danych umożliwiają zapisywanie i odczytywanie informacji w różnych standardach. Dotyczy to przede wszystkim informacji zapisanej w bazie danych, natomiast nie dotyczy, lub dotyczy w niewielkim stopniu, informacji zawartej w zapytaniach, formularzach, etykietach i projektach raportów.

Z importem danych z pliku ASCII zetknąłeś się już w tym rozdziale. Spróbuj teraz wykonać eksport danych z bazy do pliku ASCII. W programie TIG spróbuj wysłać informację w takim standardzie, jaki był używany w edytorze TAG przy korespondencji seryjnej (rys. 4.27). W programie Works wybierz standard zapisu tekstowy ze znakami systemu DOS (jest to standard Latin 2).

### 6.4.2. Korespondencja seryjna

Wiele edytorów tekstów współpracuje z programami baz danych w zakresie korespondencji seryjnej (rozdz. 4.4.6). Części zmienne korespondencji umieszcza się w bazie danych w poszczególnych polach, a edytorowi wskazuje nazwę bazy danych. W samym tekście korespondencji umieszcza się odsyłacze do nazw pól bazy danych.

Edytor TAG akceptuje części zmienne w pliku tekstowym. Każda z części zmiennych musi być wtedy poprzedzona nazwą pola (rys. 4.27). Części zmienne mogą też być przygotowane w programie dBase. Wtedy zapisuje się jedynie treść części zmiennych (nazw pól nie trzeba powtarzać wielokrotnie).

Edytor Works wymaga przygotowania danych w postaci bazy danych (rys. 4.28) i wskazania jej nazwy.

Jeżeli używany w Twojej szkole program zarządzania bazami danych współpracuje z edytorem tekstów w zakresie korespondencji seryjnej, to wykonaj następujące ćwiczenie.

### Ćwiczenie 6-20

- A. Utwórz bazę danych o polach: Imię, Ulica, Nazwisko, Miasto. Zapisz w niej dane dwóch lub trzech osób (w polu Imię umieść imię i nazwisko, a w polu Miasto — także kod pocztowy).

Jeżeli utworzyłeś już bazę danych mającą takie pola (może mieć ich więcej), to możesz ją wykorzystać. Możesz też utworzyć z niej nową (mniejszą) bazę kopiując do niej kilka rekordów.

- B. Bazę danych zapisz na dysku. (Jeżeli chcesz nawiązać do sytuacji przedstawionej na rys. 4.28, to nazwij ją DANE3.)
- C. W programie edytora przygotuj tekst korespondencji z odwołaniem się do Twojej bazy danych. Może to być np. list do czytelników biblioteki lub do absolwentów szkoły.
- D. Sprawdź, czy korespondencja seryjna jest tworzona poprawnie (skorzystaj z podglądu wydruku).

### 6.4.3. Powiązane bazy danych

Do tej pory operowaliśmy bazami danych w postaci pojedynczych tabel. Czasem wygodnie jest operować wieloma takimi tabelami jednocześnie. Na przykład w bibliotece publicznej trzeba zapisać także adresy czytelników. Można przeznaczyć na to dodatkowe pole w tabeli, jednak zapisywanie adresu tego samego czytelnika za każdym razem przy wypożyczeniu kolejnej książki nie jest wygodne. Wydaje się nawet zbędne. Ponadto powoduje zajęcie większego obszaru pamięci. W razie zmiany adresu przez czytelnika wymiana adresu na nowy jest pracochłonna. Co więcej grozi powstaniem sprzecznej informacji, jeżeli nie we wszystkich polach jednocześnie wymieni się informację na aktualną (przy wymianie automatycznie dodatkowa spacja, niewidoczna na ekranie, może spowodować, że informacja w danym polu nie zostanie wymieniona na nową).

W takiej i podobnych sytuacjach wygodnie jest zapisać dane o czytelnikach w oddzielnej tabeli — bazie danych czytelników, a w bazie danych książek odwoływać się do czytelników, wskazując ich na przykład za pomocą przydzielonych im numerów identyfikacyjnych (lub w inny jednoznaczny sposób).

Programy obsługi baz danych takie jak dBase i FoxPro umożliwiają operowanie danymi z wielu tabel jednocześnie, z wykorzystaniem logicznych związków między nimi. Efekt jest taki, jak gdyby użytkownik miał do czynienia z inną tabelą. Dotyczy to zarówno przedstawiania danych na ekranie, jak i ich wyszukiwania, porządkowania itp.

Na przykład dwie tabele z danymi: spis książek (tabela podobna do używanej w tym rozdziale, zawierająca jednak numery czytelników zamiast imion) i spis czytelników (z ich numerami i danymi osobistymi), możemy powiązać między sobą przez wspólny element, jakim są numery identyfikacyjne czytelników (rys. 6.19).

Można to uczynić na dwa sposoby:

- 1) utworzyć nową tabelę zawierającą pola z jednej i drugiej tabeli,
- 2) określić związek między istniejącymi tabelami (bez tworzenia nowej).

Każdy z tych sposobów umożliwia wyświetlenie na ekranie tabeli zawie-

rającej informacji z obu tabel składowych: z pierwszej — tytuły książek, a z drugiej — imiona, nazwiska i adresy czytelników, którzy dane książki wypożyczyli. Zilustrujemy drugi z wymienionych sposobów na przykładzie programu dBase.

...	TYTUL	NR	C_NC	IM	C_NAZ	C_ADRES
...	Mikrokomputer IBM PC...	0	1	Adaś	Minkiewicz	Krzywa 4
...	Zeszyt w kratkę	7	4	Dorota	Brzoza	Gdańska 12
...	Nastolatki gotują	9	7	Ala	Cholewińska	Gen. Fieldorfa 5
...	Paziowie króla Zygmunta	9	9	Barbara	Koncewicz	Dworcowa 12
...	Sztuka przyjaźni	10	10	Maciek	Zajac	Kościelna 8
...	Twoje życie	9	17	Heniek	Barański	Rynek 13 m.22
...	Porwany za młodu	17	13	Janek	Mirewicz	Leśna 4

Rys. 6.19. Powiązane tabele z danymi

## Ćwiczenie 6-21

- Skopiuj plik o nazwie BAZA1 do pliku KSIAZKI. Uruchom program dBase i otwórz bazę zapisaną w pliku KSIAZKI. Wypisz zapisane w niej imiona w polu Pozyczyl i przypisz im numery (nie muszą być kolejnymi liczbami). Zmodyfikuj strukturę bazy dodając pole liczbowe Nr o szerokości 2 cyfr i wpisz w nim numery czytelników. Uwaga, typ liczbowy wymaga, żeby książkom, których nikt nie pożyzył, też przypisać numer czytelnika, np. 0 (typ znakowy umożliwiłby pozostawienie pustych pozycji). Pole Pozyczyl możesz usunąć.
- Utwórz tabelę o nazwie CZYT z polami o nazwach c\_n, c\_im, c\_naz, c\_adres; pole c\_n musi być takiego samego typu jak pole Nr w tabeli KSIAZKI. Wypełnij ją podając przepisane z tabeli książki imiona i dodając pozostałe informacje. Jeżeli pole Nr jest typu liczbowego, to zapisz też „czytelnika” o numerze 0. Podobną tabelę możesz też odczytać z pliku ASCII o nazwie CZYT1.TXT znajdującym się na załączonej dyskietce (w katalogu PLIKI\BAZY; znaki polskie w standardzie Latin 2); musisz wówczas uzgodnić numerację czytelników.
- Utwórz indeks dla tabeli CZYTEL według pola c\_n; nadaj mu nazwę NRCZYT. Masz już przygotowane wszystkie dane. Obejrzyj je poleceniem BROWSE i zamknij wszystkie pliki. Sprawdź, jakie pliki pojawiły się w katalogu.
- Wykonaj sekwencję poleceń (bez tekstów zapisanych pismem pochylonym):



SELECT 2	<i>jako pierwszy</i>
USE CZYT INDEX NRCZYT	<i>otwierasz plik CZYT</i>
SELECT 1	<i>jako drugi (aktywny)</i>
USE KSIAZKI	<i>otwierasz plik KSIAZKI</i>
SET RELATION TO NUMER INTO CZYT	<i>rozszerzasz tabelę</i>

Otwierając plik CZYT używasz indeksu NRCZYT. Rozszerzając tabelę już nie musisz wskazywać na ten indeks.

Po wykonaniu tej sekwencji poleceń program już wie, w jaki sposób sięgać do odpowiednich rekordów tabeli CZYT w trakcie przeszukiwania tabeli KSIAZKI. Pół tych nie zobaczysz jeszcze na ekranie.

- E. Wykonaj polecenie SET FIELDS TO, określając pola, które miałyby być dostępne na ekranie i wypisz poleceniem LIST tytuły książek wraz z imionami i nazwiskami osób, które je pożyczyły.

```
SET FIELDS TO TYTUL, CZYT->C_IM, CZYT->C_NAZ  
LIST TYTUL, CZYT->C_IM, CZYT->C_NAZ FOR NUMER <> 0
```

Sprawdź, czy rzeczywiście na ekranie pojawiają się dane pochodzące z obu tabel: aktywnej i dołączonej. Sprawdź, jakie informacje są pokazywane poleceniem BROWSE: czy tylko z tabeli aktywnej czy też z obu tabel.

- F. Wykonaj podobne polecenie LIST z warunkiem odpowiadającym konkretnemu czytelnikowi, np. **Numer = 7**.
- G. Zakończ pracę z programem dBase i sprawdź, czy w katalogu pojawił się jakiś nowy plik.

Jeżeli w pierwszej tabeli zostałyby dodane pole daty zwrotu książki, to można by wybrać dane z obu tabel w taki sposób, by odnosiły się tylko do książek, których terminy zwrotu już minęły. Dane te mogą być uporządkowane oraz zgrupowane według czytelników. W rezultacie otrzymuje się w ten sposób wykaz czytelników, którzy nie oddali książek w terminie, wraz z wykazami książek, które każdy z nich powinien oddać. Można by wtedy wydrukować listy z przypomnieniem o wypożyczonych książkach oraz nalepki na koperty z adresami tych czytelników.

#### 6.4.4. Modele baz danych

##### Relacje

Bazy danych omawiane w książce są określane mianem **relacyjnych baz danych**. Pojęcie **relacji** pochodzi z matematyki, gdzie oznacza podzbiór iloczynu kartezjańskiego dwóch zbiorów. Iloczyn kartezjański jest to zbiór

zawierający wszystkie możliwe pary elementów z tych zbiorów. Jeżeli pierwszy zbiór zawiera wszystkie imiona uczennic i uczniów z Twojej klasy, a drugi wszystkie nazwiska, to iloczyn kartezjański składa się z wszystkich par: (imię,nazwisko). Dla 16 imion i 20 nazwisk iloczyn zawiera 320 takich par. Relacja, jako podzbiór, zawiera niektóre. Lista Twojej klasy jest zatem relacją. Tabela służy do przedstawiania relacji: kolumny tabeli odpowiadają poszczególnym zbiorom, a wiersze poszczególnym elementom relacji.

W teorii baz danych utożsamia się tabelę z relacją. Używa się pojęcia relacji  $n$ -elementowej (jako podzbioru iloczynu kartezjańskiego ( $n$ ) zbiorów); obrazem tych relacji są  $n$ -elementowe tabele. Kolumny tych tabel określa się niekiedy mianem **atrybutów relacji**, wiersze zaś nazywa się **krotkami**.

Relacja służy tu jako **model danych**. Struktura danych ma postać relacji (czyli tabel); z ich użyciem przedstawia się zarówno same dane, jak i związki logiczne między nimi. Na przykład na liście Twojej klasy wyrażony jest związek między imionami i nazwiskami, nie są to bowiem wszystkie możliwe ich kombinacje, lecz jedynie te pary, którym odpowiadają konkretne osoby (niektóre pary mogłyby nawet być nielogiczne, na przykład Anna Kowalski).

**Relacyjne bazy danych** są to bazy danych, w których dane i związki między nimi przedstawia się w postaci relacji. Ten model danych umożliwił zastosowanie formalnego aparatu matematycznego do analizy struktur danych i operacji wykonywanych na danych (takich jak na przykład łączenie tabel, czyli relacji, przedstawione w rozdziale 6.4.3). Dzięki temu można było sformułować zasady tworzenia baz danych w sposób zapewniający przedstawienie wszystkich związków logicznych między danymi (ich atrybutami), z uniknięciem zbędnego nadmiaru. Dobre podstawy teoretyczne przyczyniły się ponadto do rozwoju **programów zarządzania relacyjnymi bazami danych**, zwanych czasem **systemami** (zamiast programami).

Zyskały one popularność dzięki temu, że dla wielu użytkowników model danych w postaci tabeli jest przejrzysty. Do rozpowszechnionych w Polsce programów zarządzania relacyjnymi bazami danych należą między innymi: dBase, FoxPro, Clipper, Access. Programy takie jak Works i TIG nie umożliwiają łączenia tabel i innych operacji wykonywanych na kilku tabelach jednocześnie, niemniej do wielu zastosowań praktycznych wystarczają.

Gdybyś miał tworzyć relacyjną bazę danych do stosowania w praktyce, to wiedz, że masz dostępną pomoc w postaci zaleceń co do sposobu jej tworzenia (nie musisz polegać wyłącznie na swojej intuicji), a także podstawy teoretyczne do opisanie i zrozumienia operacji wykonywanych na danych.

## Hierarchie

Przy wyszukiwaniu książek w katalogu korzysta się niekiedy z katalogu rzeczowego, w którym książki są podzielone na kategorie. Podział ten jest wielostopniowy. Mógłbyś na przykład podzielić książki znajdujące się w Twoim domu czy bibliotece na takie kategorie, jak literatura piękna, naukowe, dziecięce; książki objęte pierwszym hasłem mógłbyś następnie podzielić na prozę i poezję, a naukowe według dziedzin, np. na nauki ścisłe, przyrodnicze i humanistyczne oraz np. encyklopedie i leksykony o szerokim zakresie haseł. W obrębie nauk ścisłych mógłbyś jeszcze dokonać podziału np. na fizykę, matematykę, chemię.

Tego rodzaju podział hierarchiczny powinien znaleźć odzwierciedlenie w Twojej bazie danych. Są programy baz danych przystosowane do uwzględniania podziałów hierarchicznych; należy do nich TIG. Do zaznaczania takich podziałów służą w nim **klasyfikatory**: hierarchiczne i równorzędne.

W programach opartych na tworzeniu relacji między tabelami danych, jak dBase i FoxPro, można również przedstawić zależności typu hierarchicznego. Możliwość przedstawiania w bazie danych zależności typu hierarchicznego mają także niektóre (zaawansowane) programy arkuszy kalkulacyjnych.

## Hipertekstowe bazy danych

Wiele programów użytkowych (nie tylko baz danych) oferuje użytkownikowi pomoc w postaci tekstu z opisanymi hasłami, przy czym niektóre hasła są wyróżnione w taki sposób, że użytkownik może wskazać je i przejść do tej części tekstu, która dotyczy danego hasła. Nie musi przy tym przechodzić hierarchicznych menu, ani nawet zastanawiać się, w jaki sposób informacja jest w tej swoistej bazie danych zorganizowana. Tego rodzaju pomoce stanowią przykład zastosowań baz danych zwanych bazami hipertekstowymi. Baz tych nie będziemy tu omawiać, a informacja o nich jest podana po to, żebyś nie odczuwał niepokoju zetknąwszy się z tą nazwą.

## Ćwiczenie 6-22

Rozważ zestaw haseł z materiału jednej lub kilku lekcji (np. matematyki, geografii lub elementów informatyki) składający się na informację, którą chciałbyś przygotować jako pomoc dla koleżanki lub kolegi, którzy byli nieobecni. Zacznij od kilku lub co najwyżej kilkunastu haseł. Zastanów się teraz, jak te hasła chciałbyś powiązać, tzn. przy każdym z haseł określ grupę innych haseł, do których z danego hasła warto byłoby móc przejść bezpośrednio.



#### 6.4.5. Własne instrukcje i programy operowania na danych

Programy zarządzania relacyjnymi bazami danych dBase i FoxPro mają bardzo rozbudowane możliwości tworzenia własnych poleceń. To już nie tylko makropolecenia, ale całe programy, mogące zawierać setki i tysiące instrukcji, realizujące złożone operacje na danych. Teksty takich programów są zapisywane w plikach. Z tą chwilą stają się one jak gdyby dodatkowymi poleceniami, tyle że wywoływanymi nie samą nazwą, lecz poleceniem *DO nazwa pliku* (bez rozszerzenia). Za pomocą takich programów można ułatwić wykonywanie typowych czynności przy obsłudze bazy danych.

Na przykład można napisać program wykonujący wszystkie czynności we wszystkich tabelach z danymi związane ze zwrotem książki przez czytelnika. Inny program realizowałby wszystkie czynności związane z wypożyczeniem książki; inny mógłby ułatwiać wyszukiwanie informacji, np. podawać wszystkie książki znajdujące się u danego czytelnika; inny wyszukiwałby osoby, u których znajduje się poszukiwana przez czytelnika książka oraz podawał przewidywany czas zwrotu.

Podobne możliwości programowania są dostępne w programach obsługi baz danych opracowanych dla środowiska Windows, np. Access (także FoxPro w wersji dla Windows) i wiele innych.

Programy takie mogą czynić obsługę bazy łatwiejszą, niż wtedy gdy operuje się nią bezpośrednio za pomocą poleceń.

Trzeba jeszcze wspomnieć o **języku zapytań SQL**, który ma walor uniwersalności, ponieważ bywa stosowany w odniesieniu do baz danych przygotowanych przez różne programy zarządzania nimi, i nawet funkcjonujące pod różnymi systemami operacyjnymi (m.in. UNIX).

Do możliwości tych powrócimy w dalszych rozdziałach.

## Pytania

1. Jakie są typy pól w używanym przez Ciebie programie zarządzania bazami danych?
2. Czy w nazwach pól możesz stosować polskie litery?
3. W jaki sposób jest zapisywana data?
4. Czym jest kreator? Czy stosowałeś go do tworzenia bazy?
5. W jaki sposób możesz przeglądać bazę?
6. Jak modyfikujesz dane?
7. Jakimi poleceniami modyfikujesz strukturę bazy?
8. Jak przebiega operacja usuwania rekordów z bazy?

9. Jakie rodzaje warunków możesz formułować przy wyszukiwaniu rekordów w bazie?
10. W jaki sposób możesz uporządkować informację w bazie?
11. Opisz tworzenie (prostego) raportu.
12. Jak możesz pogrupować informację w raporcie?
13. W jaki sposób możesz przedstawić na ekranie tylko niektóre pola bazy?
14. Po co tworzy się formularze?
15. Opisz tworzenie (prostego) formularza.
16. Czym różni się etykieta od formularza?
17. W jaki sposób użyjesz informacji z Twojej bazy danych do korespondencji seryjnej?