
Systemy zarządzania bazą danych

Spis treści

- [Ogólne pojęcie bazy danych: rekord, pole, klucz, struktura, tworzenie i aktualizacja bazy danych, bezpieczeństwo i ochrona informacji](#)
 - [Systemy bazodanowe, relacyjne bazy danych](#)
 - [Baza danych DBASE/Clipper - tworzenie, modyfikowanie, porządkowanie bazy danych](#)
-

[Ogólne pojęcie bazy danych: rekord, pole, klucz, struktura, tworzenie i aktualizacja bazy danych, bezpieczeństwo i ochrona informacji](#)

Definicje podstawowe

Baza danych

Uporządkowany zbiór danych przechowywanych w urządzeniach pamięciowych systemu komputerowego, najczęściej na dyskach magnetycznych..

Zbiór wzajemnie powiązanych ze sobą informacji, przechowywanych w pamięci zewnętrznej komputera, służących w sposób optymalny jednemu lub wielu zastosowaniom.

Baza danych powinna umożliwić szybki dostęp do dużej ilości informacji, ich modyfikację a także przeszukiwanie danych spełniających określone warunki.

System zarządzania bazą danych

program zarządzający dostępem do bazy danych

Relacja

podstawowa forma organizacji danych w bazie - zbiór rekordów (krotek)

Atrybut **A** - lub **pole** albo **kolumna**

rodzaj danych, oznaczany przez **A** (np. **Nazwisko** lub Imię lub **Wiek**).

Z każdym atrybutem jest związany zbiór wartości, np. z atrybutem Wiek związany zbiór liczb z zakresu od 0 do 110,

z atrybutem Nazwisko zbiór łańcuchów znaków (napisów) o długości do 30 znaków.

Domena atrybutu **dom(A)** lub dziedzina atrybutu

zbiór wszystkich możliwych wartości atrybutu A, np dla atrybutu **wiek** dopuszczalny zakres **0..110**

Schemat relacji **R(A₁...A_n)**

zbiór wybranych atrybutów **A₁... A_n**

Przykładowy schemat relacji PRACOWNICY: **PRACOWNICY(Nazwisko, Imię, Wiek, Pensja)**

Krotka **t** - lub **rekord**

ciąg wartości jego atrybutów, oznaczany przez t.

Przykładową krotką jest **<Kowalski, Jan, 35, 12000>**

Krotką t dla danego schematu relacji $R(A_1..A_n)$ nazywamy ciąg wartości **<a₁,...,a_n>**, takich że,

a₁ należy do dom(A₁), ... a_n należy do dom(A_n)

Relacja

skończony zbiór krotek (rekordów)

$r(A_1, \dots, A_n) = \{ \langle a_1, \dots, a_n \rangle : a_1 \in \text{dom}(A_1), \dots, a_n \in \text{dom}(A_n) \}$

gdzie $r(A_1, \dots, A_n)$ jest relacją określoną na schemacie relacji $R(A_1, \dots, A_n)$

Przykładową relacją jest

{<Kowalski, Jan, 35, 1200>, <Nowak, Piotr, 36, 2000>, <Zielińska, Anna, 25, 1400>}

Krotki relacji nie są uporządkowane, można zmieniać ich kolejność.

Klucz relacji

taki zbiór identyfikujący relacji, którego żaden podzbiór nie jest zbiorem identyfikującym relacji

Wyróżnia się klucze **proste i złożone**.

Klucz jest **kluczem prostym**, jeśli zbiór identyfikujący relacji jest zbiorem **jednoelementowym**, w przeciwnym razie klucz jest **złożony**.

W ogólności, w relacji można wyróżnić wiele kluczy, które nazywamy kluczami **potencjalnymi**.

Wybrany klucz spośród kluczy potencjalnych nazywamy **kluczem głównym**.

Struktura

Dane w bazie danych są wprowadzane i zapamiętywane w formie strukturalnej.

Elementarna struktura w bazie danych nazywa się **rekordem** (record) lub **krotką**.

Zbiór informacji, zwany *plikiem* (file) składa się z **rekordów**.

Rekord z kolei złożony jest z **pól** (field) - *atrybutów*.

Bezpieczeństwo

W bazach wielodostępnych musi być stosowany mechanizm blokowania rekordów (record locking), polegający na tym, że gdy jeden użytkownik zmienia rekord, musi on być zablokowany (do czytania lub modyfikacji) dla innych użytkowników. Stosowane są też zabezpieczenia w postaci haseł oraz udostępniania dla wybranych użytkowników tylko określonych pól.

Tablice

najczęstszy zapis relacji - **wiersze tablic** odpowiadają **krotkom**, a **kolumny atrybutom**

Rozróżnia się 2 podstawowe typy baz danych

- **kartotekowe bazy danych**
- **relacyjne bazy danych**

Wprowadzenie, przeznaczenie baz danych

Jednym z najważniejszych zastosowań komputerów są systemy baz danych.

Najogólniej przez **bazę danych** rozumiemy **uporządkowany zbiór danych przechowywanych w pamięci zewnętrznej komputera,**

a przez **system bazy danych** - **bazę danych wraz ze środkami programowymi umożliwiającymi operowanie na niej, np. wyszukiwanie i aktualizowanie informacji.**

W nowoczesnym przedsiębiorstwie nie można sobie wyobrazić dobrze funkcjonującego przedsiębiorstwa ani biura, którego działalność mogłaby być prowadzona bez korzystania z systemów baz danych.

Upowszechnienie systemów baz danych było związane z gwałtownym i masowym zastosowaniem mikrokomputerów.

Jest wiele pakietów określanych "mikrokomputerowy system bazy danych".

Spośród nich dużą popularność osiągnął system dBase, np. dBase III firmy Ashton-Tate z USA. dBase III + jest wielodostępną, sieciową wersją systemu dBase III.

System dBase III reklamowany jako "relacyjna baza danych" jest podobnie jak wiele innych systemem *quasi-relacyjnym*.

W pełni relacyjna baza danych powinna mieć 20 cech.

U podstaw relacyjnych baz danych leży teoria matematyczna, umożliwiająca zalgorytmizowanie procesu projektowania baz danych oraz opracowanie języków dostępu do danych,

z których użytkownik określa tylko które dane go interesują, a nie jak do nich dotrzeć.

Standardem w zakresie relacyjnych baz danych jest język **SQL**.

Gromadzenie danych, a zwłaszcza wyszukiwanie informacji w wielkich zbiorach danych jest pracochłonne i zabiera wiele czasu.

Do tego celu zastosowano komputery, wykorzystując ich możliwości pamiętania dużych ilości informacji i szybkiego ich przetwarzania.

Opracowano specjalne **systemy zarządzania bazą danych DBMS - Data Base Management Systems**.

Pierwotnie do tego celu stosowano duże systemy komputerowe.

Dzięki zastosowaniu mikrokomputerów zaczęto stosować bazy danych powszechnie.

Mogą być używane zarówno do opracowywania katalogów książek w bibliotece, spisów towarów w magazynie, zamówień, pracowników, telefonów, podatników, klientów, wydatków, rezerwacji miejsc i innych.

Systemy bazodanowe, relacyjne bazy danych

Systemy zarządzania bazą danych (nazywane w skrócie bazami danych) pozwalają na wprowadzenie i przechowywanie ich w pamięci (np. na dysku

Realizują one *4 najważniejsze funkcje*:

- sortowanie
- wyszukiwanie
- przetwarzanie
- wizualizację danych

*Programy do zarządzania bazami danych powinny **umożliwić** przede wszystkim:*

- tworzenie nowych baz danych
- modyfikowanie struktury bazy
- dodawanie lub kasowanie rekordów
- wyszukiwanie rekordów spełniających określone kryteria

- przeglądanie i drukowanie wybranych pól z wybranych rekordów
- porządkowanie bazy wg określonych kryteriów
- sporządzanie raportów czyli zestawień i drukowanie zbiorczych zestawień o wybranych polach
- analizowanie danych (sumowanie, obliczanie średniej itp.)

Dane w bazie danych są wprowadzane i zapamiętywane w formie strukturalnej. Elementarna struktura w bazie danych nazywa się **rekordem** (record) lub krotką. Zbiór informacji, zwany *plikiem* (file) składa się z *rekordów*. **Rekord** z kolei złożony jest z *pól* (field) - **atrybutów**.

Przykład rekordu bazy danych w dBASE

NAZWA_T OWA	TYP_TOW ARU	CENA_TO WAR	ILOSC_T OWA	DATA_D OSTA	DATA_W AZN	PRODUC ENT
JABŁKA	JONATHA N	120.00	68.00	02/05/87	02/15/87	FARMA 1

Pola w tej bazie to m.in. NAZWA_TOWA, TYP_TOWARU, itd.

Jak przebiega praca z bazą danych

Systemy zarządzania bazą danych są zwykle programami ogólnego przeznaczenia, tzn. użytkownik ma możliwość dostosowania ich do konfiguracji sprzętu oraz do wymagań danego zastosowania.

Program bazodanowy trzeba najpierw zainstalować.

Dla konkretnej bazy ustala się

- sposób wprowadzania i przechowywania danych, czyli **strukturę rekordu** - ustala pola rekordu i nadaje im nazwy,
- definiuje **formularze** do wprowadzania danych,
- definiuje się także **raporty** czyli sposoby wyświetlenia informacji

Następnym etapem pracy jest **wprowadzenie danych** - przypomina to wypełnianie formularzy.

Największą zaletą baz danych jest szybkie wyszukiwanie informacji oraz ich przetwarzanie.

Można np. wybierać tylko rekordy spełniające określone kryteria, np. opisujące towary od konkretnego producenta.

Selekcja może być wielostopniowa, tzn. wśród już wybranych rekordów można określać kolejne kryteria wyboru.

Wybrane rekordy można dalej przetwarzać lub wyświetlać.

Dane mogą być także posortowane, czyli uporządkowane w pewnej kolejności, np. w porządku alfabetycznym wg ustalonego pola.

Wiele systemów pozwala na tworzenie plików indeksowych, w których są zapisywane informacje o rozmieszczeniu pól i rekordów w bazie danych.

Korzystanie z tych informacji znacznie przyspiesza późniejsze wyszukiwanie danych.

Informacje zapisane w pliku można automatycznie przetwarzać.

Proste bazy danych pozwalają np. na obliczenie sum elementów z konkretnych pól i wartości średnich.

Złożone systemy mają własne języki programowania, w których można zaprogramować skomplikowane operacje, wykonywane na zawartości wybranych rekordów lub pól z tych rekordów.

Można np. obliczyć wartość towarów w sklepie, zmienić ceny towarów itd.

Skomplikowane bazy danych umożliwiają jednoczesne operowanie na wielu plikach i łączenie danych z różnych plików a nawet równoczesne wykorzystywanie ich przez wielu użytkowników.

Rodzaje baz danych

Bazy danych różnią się między sobą sposobem organizacji zapamiętywanych informacji i zarządzania nimi

Kartotekowe bazy danych - systemy kartotek (flat-file database, card file)

Systemy te są najprostszymi bazami danych. Plik składa się z odrębnych dokumentów o identycznej strukturze lecz o różnej zawartości.

Każdy z tych dokumentów stanowi rekord bazy danych. Plik składa się więc z rekordów

Rekordy mogą być podzielone na pola, które mogą zawierać tekst, liczby, daty oraz formuły, służące do obliczenia wartości pola w zależności od wartości innych pól.

Każdemu polu przypisuje się nazwę. Wszystkie rekordy składają się z pól o tych samych nazwach i tego samego typu, ale zawartość pól w poszczególnych rekordach może być różna. Wszystkie rekordy mają taką samą strukturę i nie są ze sobą powiązane.

Plik przypomina zbiór **kartotekę** jednakowych formularzy.

Przykładem kartotekowej bazy danych jest zbiór książkowych kart katalogowych.

Każda karta stanowi rekord, a poszczególne pozycje jak autor, tytuł itp. są polami rekordu.

Przykłady programów do obsługi kartotekowych baz danych

Nutshell Information Manager firmy Leading Edge Products. Program nie stawiał praktycznie ograniczeń na liczbę rekordów, liczbę pól i wielkość pól w rekordach.

Zdefiniowanie bazy polega na podaniu nazw i typu poszczególnych pól. Operacje wykonuje się przez wybór opcji z menu.

Poszczególne pola można dowolnie rozmieszczać na ekranie tworząc różnego rodzaju formularze.

Do określenia formy wydruku raportu służą formularze (szablony) wydruku, definiowane na ekranie.

Program służył m.in. do obsługi katalogów, list płac, kartotek personalnych w zakładzie pracy, kartotek pacjentów w szpitalach.

Wadą jest brak możliwości przeglądania raportu na ekranie przed jego wydrukiem.

Reflex firmy Borland International - pozwala w różnych przekrojach analizować bazę danych reprezentowaną przez tablicę oraz przedstawić wyniki w postaci wykresów.

Kartoteka (Cardfile) z grupy Akcesoria Windows 3.1x. Służy do prowadzenia prostej kartotekowej bazy danych.

Użytkownik tworzy bazę danych składającą się z kart (rekordów) o 2 polach: pola nagłówka (indeksu) i pola informacji.

Zawartości tych pól w bazie są na ogół różne, ale ich rozmiary są dla wszystkich kart jednakowe.

W polu informacyjnym może znajdować się zarówno tekst jak i grafika.

Mogą tu być skopiowane, wbudowane lub dołączane dowolne obiekty OLE.

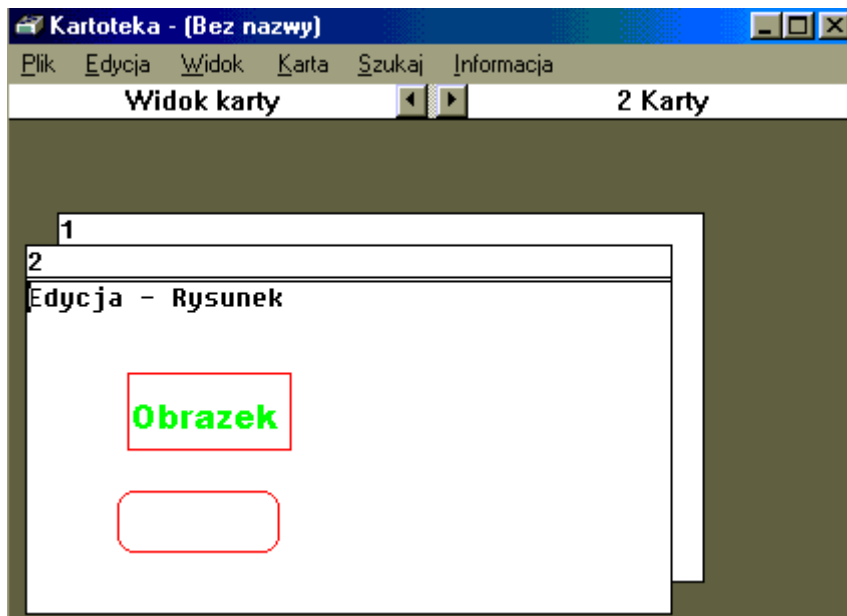
Do pola nagłówka (na rys. 1 lub 2) dostajemy się przez podwójne kliknięcie myszą na tym polu lub przez Edycja/Indeks.

Pole informacyjne dostępne jest przez jedno kliknięcie myszą w jego obrębie.

Poruszanie się po kartotece umożliwia np. kliknięcie na wystającym brzegu karty lub naciskanie na poziomą strzałkę w prawo (do przodu) lub lewo (wstecz).

Dłuższe przytrzymanie strzałki powoduje szybkie przewijanie kart. W polu informacyjnym można pisać teksty lub rysować/wstawiać rysunki.

Włączanie trybu tekstowego (domyślny) przez Edycja/Tekst, a trybu Rysunek przez Edycja/Rysunek.



TIG

Inne znane kiedyś systemy kartotek:

Nazwa	Producent	Sprzęt	System operacyjny.	max liczba znaków w polu	max liczba pól w rekordzie	max liczba rekordów w pliku	max liczba otwartych plików	rekord o stałej lub zmiennej długości	menu lub polecenia	organizacja indeksu
Master file	Campbell	ZX Spektru	system ZX	128	26	ograniczona PaO	1	stałej	menu	nie ma

		m	Spectrum			(ok. 200)				
DBMaster v.4+	Stone ware Inc.	Apple	DOS 3.0	100	100	nieograniczona	1	stałej	-	ISAM
Supebase 64	Precision Software	Commodore 64	system Commodore 64	1108	127	nieograniczona	1	stałej	-	B-drzewa
Omnifile	SSR Corp.	IBM PC XT, 2 FD	DOS	1828	64	nieograniczona	1	zmiennej	-	zmodyfikowane B-drzewa

Relacyjne bazy danych

W relacyjnych bazach danych wykorzystano teorię algebry relacji.

Pliki składają się z rekordów, zaś rekordy z pól ale różnica w stosunku do baz kartotekowych polega na tym, że baza może zawierać kilka plików (lub tabel), które to pliki (tabele) różnią się strukturą rekordów.

Pomiędzy rekordami jest określona pewna relacja porządkująca.

Możliwe jest łączenie rekordów z różnych plików, jeśli rekordy takie mają przynajmniej jedno pole wspólne.

Jeśli mamy jedną lub więcej baz zawierających przynajmniej jedno wspólne pole, wówczas może zachodzić potrzeba utworzenia nowej bazy zawierającej połączone informacje lub wydrukowania raportu z połączonymi informacjami.

Przykładowo może to dotyczyć informacji o stażu pracy z kartoteki personalnej i wynagrodzeniu z listy płac.

Wspólnym polem może być nazwisko lub nr pracownika

Innym przykładem może być baza danych prowadzona z księgowości i baza prowadzona w magazynie, gdzie wspólnym polem jest nazwa (lub indeks) zakupionych i magazynowanych materiałów.

Takie bazy nazywamy **relacyjnymi**.

W uproszczeniu można je przedstawić za pomocą tablic (nazywanych relacjami) powiązanych między sobą poprzez kolumny, w których występują wspólne dane.

Odszukanie określonego rekordu (wiersza) w jednej tablicy pozwala poprzez kolumny zawierające wspólne dane wybrać odpowiedni rekord drugiej tablicy.

Jedną dużą bazę można podzielić na mniejsze, a następnie łączyć informacje w miarę potrzeby.

Bazy relacyjne są bazami wieloplikowymi lub wielotabelowymi.

Programy zarządzania relacyjnymi bazami danych zawierają zbiór rozkazów, które umożliwiają wykonywanie operacji na całej bazie.

Zaletą relacyjnych baz danych jest uniezależnienie formy wprowadzania informacji od sposobu ich użytkowania.

Bazy pozwalają bowiem na zmiany struktury rekordu.

Do programów zarządzania relacyjnymi bazami danych należą m.in. programy serii **dBASE** (w różnych odmianach: dBase III+, dBase IV, Foxbase oraz różne wersje języka Clipper)

Inne typy (struktury) baz danych

- **Hierarchiczne** (hierarchical) - plik składa się z rekordów o stałej strukturze. Rekordy te są na stałe połączone ze sobą i tworzą hierarchiczną konstrukcję podobnie jak katalogi w systemie Unix. Bazy hierarchiczne oparte są na pionowych powiązaniach między plikami, podobnie jak to ma miejsce w strukturze katalogów i plików w systemach operacyjnych, np. DOS.
- **Sieciowe** (network) - przypominają bazy hierarchiczne, z tym, że każdy rekord może być połączony z dowolnym innym rekordem. Sieciowe bazy danych różnią się od baz hierarchicznych, że występują w nich nie tylko powiązania pionowe, ale także poziome lub skośne. Tak np. książka, w której rozdziały i podrozdziały stanowią hierarchiczną bazę danych (powiązania pionowe) może zawierać w poszczególnych rozdziałach odnośniki do innych rozdziałów (powiązania poziome)
- **O formacie swobodnym** (free format) - dane mogą być wprowadzane w dowolnej postaci: rekordów, tabel, bloków tekstu itd. Każdy element takiej bazy jest opatrzony pewnym słowem kluczowym, które jest używane przy wyszukiwaniu informacji. Bazy danych o formacie swobodnym zawierają dowolne informacje tekstowe i liczbowe bez podziału na pola. Każda informacja jest opisana słowami kluczowymi, wg których można ją następnie odszukać. Przykładem takiej bazy może być zbiór wypisów z czasopism, w których słowami kluczowymi są np. tytuł czasopisma i dziedzina, której dotyczy dany wypis. Bazę tę można traktować jako kartotekę, w której występuje kilka pól o stałej długości (słowa kluczowe) i jedno pole o dowolnej długości (tekst wypisu)
- **Wielodostępne** bazy danych (multiuser) - mogą być jednocześnie przeglądane i modyfikowane przez wielu użytkowników pracujących z różnych terminali lub komputerów połączonych w sieć. Wielodostępna baza danych może mieć różną strukturę, np. może być relacyjna, hierarchiczna lub zorganizowana w system kartotek. Np. wielodostępna, sklepowa baza danych może obsługiwać sieć sklepów. Dostęp do takich baz danych możliwy jest również telefonicznie za pomocą modemu, np. w USA popularna była baza Dow Jones, dla przedsiębiorców. W bazach wielodostępnych musi być stosowany mechanizm blokowania rekordów (record locking), polegający na tym, że gdy jeden użytkownik zmienia rekord, musi on być do czytania lub modyfikacji dla innych użytkowników. Stosowane są też zabezpieczenia w postaci haseł oraz udostępniania dla wybranych użytkowników tylko określonych pól. Spośród programów obsługi wielodostępnych baz danych, które mogą być stosowane w sieciach lokalnych, można wymienić **dBASE III Plus** i nowsze firmy Ashton-Tate,

Network Revelation (Cosmos), R:base System V (Microrim).
Program Smart Data Base Manager LAN Multi-user Version (Innovative Software) stosowany był często do porównywania jakości sieci lokalnych.
Dużą popularnością cieszy się program **Informix** (Relational Database Systems), wykorzystujący język **SQL** (Standard Query Language), oferowany w wersjach pracujących pod systemem DOS i Unix (Xenix).

Pojęcia podstawowe

System bazy danych składa się z bazy danych i systemu zarządzania bazą danych.

Baza danych jest to uporządkowany zbiór danych przechowywanych w urządzeniach pamięciowych systemu komputerowego, najczęściej na dyskach magnetycznych.

System zarządzania bazą danych jest programem zarządzającym dostępem do bazy danych.

Podstawową formą organizacji danych w bazie jest **relacja**.

Rodzaj danych (np. **Nazwisko**, **Wiek**, **Pensja**) w teorii relacyjnych baz danych nazywa się **atrybutem** i jest oznaczany przez **A**.

Z każdym atrybutem jest związany zbiór dopuszczalnych wartości (np. Wiek - liczby 0..110, Nazwisko - łańcuchy znaków o długości do 30).

Zbiór wszystkich możliwych wartości atrybutu A nazywamy jego **domeną (dziedziną)** i oznaczamy przez **dom(A)**.

Zbiór wybranych atrybutów A1, ..., An nazywamy **schematem relacji** i oznaczamy **R(A1,...,An)**.

Przykład: schematem relacji PRACOWICY jest **PRACOWNICY (Nazwisko, Imię, Wiek, Pensja)**.

Dla danego schematu relacji ciąg wartości jego atrybutów nazywamy **krotką (rekordem)** i oznaczamy przez **t**.

Przykładową krotką jest <Kowalski, Jan, 35, 2000>.

Innymi słowy, dla danego schematu relacji R(A1,...,An),

krotką t nazywamy ciąg wartości <a1,...,an>,

takich, że a1 (- dom(A1),..., an (- dom(An)

Relacja określona na danym schemacie relacji nazywamy **skończony zbiór krotek (rekordów)**.

Formalnie

$r(A1,...,An) = \{ \langle a1,...,an \rangle : a1(-dom(A1),...an(-dom(An)) \}$, gdzie $r(A1,...,An)$ jest relacją określoną na schemacie relacji $R(A1,...,An)$.

Przykładową relacją jest {<Kowalski, Jan, 35, 2000>, <Nowak, Piotr, 36, 1500>, <Zielińska, Anna, 25, 1000<}

Krotki relacji nie są uporządkowane. *Można dowolnie zmieniać kolejność krotek, bez zmiany relacji.*

Z matematycznego punktu widzenia przestawienie atrybutów powoduje zmianę relacji.

W wielu systemach danych, również w dBase III, pomija się to ograniczenie i przyjmuje, że uporządkowanie atrybutów nie jest znaczące.

Najczęściej relacje reprezentuje się w postaci **tablic**, których wiersze odpowiadają krotkom (rekordom), a kolumny atrybutom (polom).

Przykładowa relacja w postaci tablicy			
Nazwisko	Imię	Wiek	Pensja
Kowalski	Jan	35	2000
Nowak	Piotr	36	1500
Zielińska	Anna	25	1000

Operacje relacyjne

- selekcja
- projekcja
- połączenie
- suma relacji
- różnica relacji
- iloczyn kartezjański

Na 3 pierwszych operacjach: selekcji, projekcji i połączeniu opiera się tzw. **język manipulowania danymi**, umożliwiający użytkownikom baz danych dokonywanie przekształceń relacji, wynikających z potrzeb.

Zazwyczaj użytkownicy żądają dostępu do innych relacji niż te, które są przechowywane. Zrealizowanie takiego żądania wymaga przekształcenia relacji przechowywanych w bazie danych w relacje określone w żądaniu.

Tak utworzone relacje mogą być "pionowymi" lub "poziomymi" podzbiorami innych relacji, tzn. relacjami powstałymi przez usunięcie niektórych atrybutów lub krotek, bądź mogą być wynikiem połączenia różnych relacji w jedną.

Operacja projekcji

Operacją projekcji relacji $r(A_1, \dots, A_n)$ na zbiorze atrybutów $X=(A_i, \dots, A_j)$, co zapisujemy Π_x $r(A_1, \dots, A_n)$, nazywamy przekształcenie relacji $r(A_1, \dots, A_n)$ w relację wynikową postaci:
 $\Pi_x r(A_1, \dots, A_n) = \{t[X] : t \in r(A_1, \dots, A_n)\}$

Operacja projekcji umożliwia utworzenie "pionowego" podzbioru relacji przez wybór określonych atrybutów (pól).

Relacja KIEROWNICY_DZIALOW				
Id_Kierownika	Nazwisko	Id_Dzialu	Zarobki	Adres_Dzialu
1	Wałęcki	1	2500	Poznań, ul. Strzelecka 5
2	Kościański	2	1800	Pniewy, ul. Prosta 3
6	Pietras	3	2300	Poznań, ul. Kórnicka 30

Wynik operacji projekcji $\Pi_{\text{Id_Kierownika, Nazwisko, Zarobki}}$
KIEROWNICY_DZIALOW:

Wynik projekcji relacji KIEROWNICY_DZIALOW		
Id_Kierownika	Nazwisko	Zarobki
1	Wałeccki	2500
2	Kościański	1800
6	Pietras	2300

W systemach zarządzania danymi operacja ta jest formułowana za pomocą poleceń manipulowania danymi.

Np. w dBasee służą do tego polecenia LIST i DISPLAY.

Operacja selekcji

Operacją **selekcji** relacji $r(A_1, \dots, A_n)$ względem kryterium selekcji E , co zapisujemy $\sigma_E r(A_1, \dots, A_n)$, nazywamy przekształcenie relacji $r(A_1, \dots, A_n)$ w relację wynikową postaci $\sigma_E r(A_1, \dots, A_n) = \{t : t \in r(A_1, \dots, A_n) \wedge E(t) = \text{prawda}\}$

Operacja selekcji umożliwia utworzenie "poziomego" podzbioru relacji przez wybór krotek (rekordów) spełniających określony warunek.

Wynik operacji selekcji $\sigma_{\text{Id_Dzialu} \leq 3 \text{ .AND. Zarobki} \geq 2000}$ KIEROWNICY_DZIALOW

Id_Kierownika	Nazwisko	Id_Dzialu	Zarobki	Adres_Dzialu
1	Wałeccki	1	2500	Poznań, ul. Strzelecka 5
6	Pietras	3	2300	Poznań, ul. Kórnicka 30

W języku naturalnym można tę operację sformułować następująco:

"Z relacji KIEROWNICY_DZIALOW wybierz informacje o kierownikach tych działów, których identyfikatory są mniejsze lub równe 3 i zarobki są większe równe 2000".

Operacja połączenia

Operacja **połączenia** polega na scaleniu odpowiednich krotek (rekordów) 2 różnych relacji pod warunkiem spełnienia warunku logicznego θ nałożonego na atrybuty połączeniowe.

Np. może być operacja połączenia PRACOWNICY θ WYDZIALY, gdzie θ ma postać: $\text{Id_Wydzialu} = \text{Id_Wydzialu}$

Projektowanie bazy danych

Punktem wyjścia są relacje wymagane przez użytkowników a właściwie informacje zebrane przez projektanta o wymaganych atrybutach bazy danych, celem jest natomiast zdefiniowanie schematów relacji, które mają być przechowywane w bazie danych.

Trzeba tak zaprojektować rozdział zbioru atrybutów na schematy relacji, aby uzyskać pewne pożądane właściwości bazy danych.

Niewłaściwe zaprojektowanie schematów relacji może być przyczyną dublowania się danych, ich niespójności i anomalii podczas ich aktualizowania.

W ramach projektowania schematów baz danych, najistotniejszą czynnością jest **normalizacja relacji**, czyli doprowadzenie relacji do odpowiedniej postaci normalnej.

Pierwsza postać normalna jest immanentną cechą relacji, gdyż wymagania tej postaci są zawarte w definicji relacji.

Postać normalizacji polega w tym przypadku na doprowadzeniu określonego zbioru danych do postaci relacji.

Mając bazę danych złożoną z relacji w pierwszej postaci normalnej, jest się narażonym na wszystkie niekorzystne zjawiska wymienione wyżej, dlatego należy doprowadzić relacje do kolejnych postaci normalnych.

Proces normalizacji polega na odpowiednim podziale relacji na mniejsze, w wyższej postaci normalnej.

Pierwsza postać normalna relacji

Relacja jest w pierwszej postaci normalnej, jeżeli każda wartość atrybutu w każdej krotce tej relacji jest wartością elementarną, czyli nierozkładalną.

Z definicji pierwszej postaci normalnej relacji wynika, że każdemu elementowi relacji znajdującemu się na przecięciu dowolnej krotki i dowolnego atrybutu odpowiada pojedyncza wartość, a nie zbiór wartości.

Relacja nieznormalizowana						
Nr_Zamow	Id_Dostawcy	Nazwa_Dostawcy	Adres_Dostawcy	Id_Czesci	Nazwa_Czesci	Ilosc
001	300	FSO	Warszawa	53	gaźnik	100
				57	wał	50
				59	łotnik	500
002	400	FSM	Tychy	54	gaźnik	200

Relacja znormalizowana						
Nr_Zamow	Id_Dostawcy	Nazwa_Dostawcy	Adres_Dostawcy	Id_Czesci	Nazwa_Czesci	Ilosc
001	300	FSO	Warszawa	53	gaźnik	100
001	300	FSO	Warszawa	57	wał	50
001	300	FSO	Warszawa	59	łotnik	500
002	400	FSM	Tychy	54	gaźnik	200

Druga postać normalna

Aby przedstawić 2-gą postać normalną należy wprowadzić kilka pojęć

Zbiór identyfikujący relacji

taki zbiór atrybutów tej relacji, których kombinacje wartości jednoznacznie identyfikują każdą krotkę relacji

Klucz relacji

taki zbiór identyfikujący relacji, którego żaden podzbiór nie jest zbiorem identyfikującym relacji

Wyróżnia się klucze *proste* i *złożone*. Klucz jest kluczem prostym, jeśli zbiór identyfikujący relacji jest zbiorem jednoelementowym, w przeciwnym razie klucz jest złożony.

W ogólności, w relacji można wyróżnić wiele kluczy, które nazywamy kluczami *potencjalnymi*. Wybrany klucz spośród kluczy potencjalnych nazywamy *kluczem głównym*.

Zależność funkcjonalna

Atrybut B relacji r jest funkcjonalnie zależny od atrybutu A tej relacji, jeśli zawsze każdej wartości a atrybutu A odpowiada nie więcej niż jedna wartość atrybutu B. Inaczej A identyfikuje B.

Np. Id_Dostawcy jest funkcjonalnie zależny od atrybutu Nr_Zamowienia, gdyż jedno zamówienie jest kierowane tylko do jednego dostawcy, a zatem numer zamówienia jednoznacznie określa identyfikator dostawcy. Odwrotna zależność nie jest prawdziwa.

Pełna zależność funkcjonalna

Atrybut B relacji r jest w pełni funkcjonalnie zależny od zbioru atrybutów X, jeśli jest funkcjonalnie zależny od niego, ale nie jest funkcjonalnie zależny od żadnego podzbioru zbioru X.

Druga postać normalna:

Dana relacja jest w drugiej postaci normalnej, jeśli każdy atrybut tej relacji nie wchodzący w skład żadnego klucza potencjalnego jest w pełni funkcjonalnie zależny od wszystkich kluczy potencjalnych.

W celu uzyskania drugiej postaci normalnej należy podzielić relację na zbiór takich relacji, których wszystkie atrybuty będą w pełni funkcjonalnie zależne od kluczy.

Przechodnia zależność funkcjonalna

Niech X, Y i Z będą 3 rozłącznymi podzbiórami atrybutów danej relacji. Podzbiór atrybutów Z jest przechodnio funkcjonalnie zależny od podzbioru atrybutów X, jeśli podzbiór atrybutów Z jest funkcjonalnie zależny od podzbioru atrybutów Y, podzbiór atrybutów Y jest funkcjonalnie zależny od podzbioru atrybutów X, natomiast podzbiór atrybutów X nie jest funkcjonalnie zależny od Y lub podzbiór atrybutów Y nie jest funkcjonalnie zależny od Z.

Trzecia postać normalna relacji

Dana relacja jest w trzeciej postaci normalnej, jeśli jest ona w drugiej postaci normalnej i każdy jej atrybut nie wchodzący w skład żadnego klucza

potencjalnego nie jest przechodnio funkcjonalnie zależny od żadnego klucza potencjalnego tej relacji.

Aby uzyskać trzecią postać normalną relacji, której atrybuty pozostają w przechodniej zależności funkcjonalnej, należy podzielić ją na 2 relacje.

Wyróżnia się jeszcze czwartą i piątą postać normalną relacji

Baza danych dBASE/Clipper - tworzenie, modyfikowanie, porządkowanie bazy danych

Charakterystyka ogólna dBASE, typy pól

Program bazy danych **dBASE** firmy Ashton Tate, działający na komputerze IBM PC był i jest jeszcze jednym z najpopularniejszych systemów do zakładania, przechowywania i przetwarzania dużych zbiorów informacji.

System dBASE II został opracowany dla mikrokomputerów 8-bitowych, pracujących z systemem operacyjnym CP/M.

Dużą popularność zdobył potem **dBASE III** na komputery IBM PC, zwłaszcza **dBASE III Plus**, umożliwiający dostęp do plików za pośrednictwem sieci.

System dBASE III umożliwia tworzenie tzw. relacyjnej bazy danych, gdzie podstawowymi składnikami bazy są **relacje**.

Podejście relacyjne do danych wynika z faktu, że zawartość plików spełniających pewne warunki można traktować jako relacje matematyczne, do których stosuje się elementarną teorię relacji.

Każdą relację w systemie bazy danych możemy przedstawić w postaci tablicy zawierającej szereg wierszy podzielonych na określoną liczbę kolumn.

Zamiast określenia relacja używa się często określenia plik bazy danych, wiersze tablicy nazywa się **rekordami**, a jej kolumny **polami**.

Pliki bazy danych, zawierające relacje mają rozszerzenie **DBF**.

W przypadku dBASE III baza danych może tworzyć wiele plików, przy czym jednocześnie otwartych może być najwyżej 9.

Każdy plik składa się z nagłówka, w którym określona jest liczba rekordów i nazwy oraz charakterystyki pól tworzących rekord, oraz z rekordów zawierających dane.

Każdy rekord w pliku ma tę samą strukturę, określoną podczas zakładania pliku bazy danych - długość rekordów i pól jest stała, bez względu na długość zapisywanej informacji.

Typ pola w rekordzie opisuje charakter danej, jaka będzie przechowywana w polu.

System dBASE III akceptuje dane 5 podstawowych typów: dane **tekstowe** - charakter field, dane **liczbowe** - numeric field, dane **logiczne** - logical field,

dane w postaci **daty** - date field, **wskaźnik** do tzw. **notatki** (pola memo).

Notatki zapisuje się do oddzielnego pliku o rozszerzeniu **DBT**.

Typy pól w DBase			
Oznaczenie pola	Typ pola	Długość	Opis
C	znakowe (tekstowe), Charakter	254 znaki j. Clipper do 32 KB	dowolny ciąg znaków - napisy złożone z liter, cyfr i innych znaków dostępnych na klawiaturze
N	numeryczne (liczbowe), Numeric	19 znaków, j. Clipper do 30 zn, DBase IV do 20 zn	cyfry, znak + lub -, kropka dziesiętna
L	logiczne, logical	1 znak	1 znak: t, T, y, Y - jeśli prawda lub n, N, f, F jeśli fałsz
D	data, date	8 znaków	data w postaci mm/dd/rr lub w wersji polskiej w postaci rr/mm/dd
M	notatnikowe (opisowe) memo	10 znaków	wskaźnik do zbioru .DBT
F	<i>numeryczne, float</i> <i>tylko w DBase IV</i>	20 znaków	<i>liczby zmiennoprzecinkowe, inna reprezentacja wewnętrzna niż w typie N</i>

Liczba rekordów w zbiorze max 1 miliard. Długość rekordu max. 4000 bajtów, max. 128 pól. W DBase IV max liczba pól w rekordzie 255, dług. rekordu też 4000 bajtów, w j. Clipper 5.0 mogą wystąpić max. 1024 pola.

W dBase III+ w zbiorze .dbt (memo) treść pola **memo** wynosi max. 5000 znaków, w dBase IV jest dowolna (ograniczona dostępną pamięcią), w j. Clipper 5.0 rozmiar pola memo ograniczony jest do 64 KB.

Poza plikami bazy danych, zawierającymi relacje (pliki o rozszerzeniu **DBF**) i plikami notatek (**DBT**), w systemie istnieją inne typy plików, zawierające informacje wspomagające działanie programu. Największe znaczenie mają pliki indeksowe o nazwach określonych przez użytkownika i rozszerzeniach **NDX**.

Pliki o rozszerzeniu **MEM** zawierają dane i zmienne, trzymane podczas działania programu w pamięci operacyjnej.

Pliki **FMT** zawierają szablony, wg których wyświetlana jest informacja z bazy na ekranie monitora.

Pliki **FRM** zawierają schematy tworzenia raportów przez drukarki.

Pliki indeksowe mają za zadanie skrócenie czasu przeszukiwania pliku bazy danych. Plik indeksowy umożliwia uszeregowanie rekordów wg informacji zawartych w określonym polu. Założenie pliku indeksowego nie powoduje uporządkowania rekordów w pliku bazy danych (DBF), lecz jej uporządkowanie w ramach pliku indeksowego (NDX).

W pliku indeksowym umieszcza się zawartość wybranego pola i nr rekordu, w którym znajduje się pole wg kolejności alfabetycznej.

Z plikiem bazy danych może być skojarzonych wiele plików indeksowych, ale równocześnie może być otwartych nie więcej niż 7.

Proces użytkowania bazy danych można podzielić na 2 zasadnicze fazy.

Faza zakładania bazy danych, podczas której należy opisać strukturę rekordów w plikach baz danych, a następnie wpisać do plików określoną liczbę danych początkowych (założyć rekordy - wypełnić ich pola).

Od momentu założenia zaczyna się faza druga, związana z przetwarzaniem i konserwacją informacji zawartych w bazie.

Praca w dBase może odbywać się na 2 sposoby:

- w trybie **konwersacyjnym**
- w trybie z **asystentem**

W trybie **konwersacyjnym**, bezpośrednim tzw. trybie z kropką. Kropka jest standardowym znakiem zgłoszenia dBase i po niej wydaje się polecenia języka dBase, które są natychmiast wykonywane.

W trybie z **asystentem** - z wykorzystaniem programu pomocniczego (asystenta) o nazwie ASSIST w dBase III+ lub Centrum Sterowania w dBase IV.

W tym trybie użytkownik wybiera odpowiednie pozycje z menu i potwierdza klawiszem <Enter>.

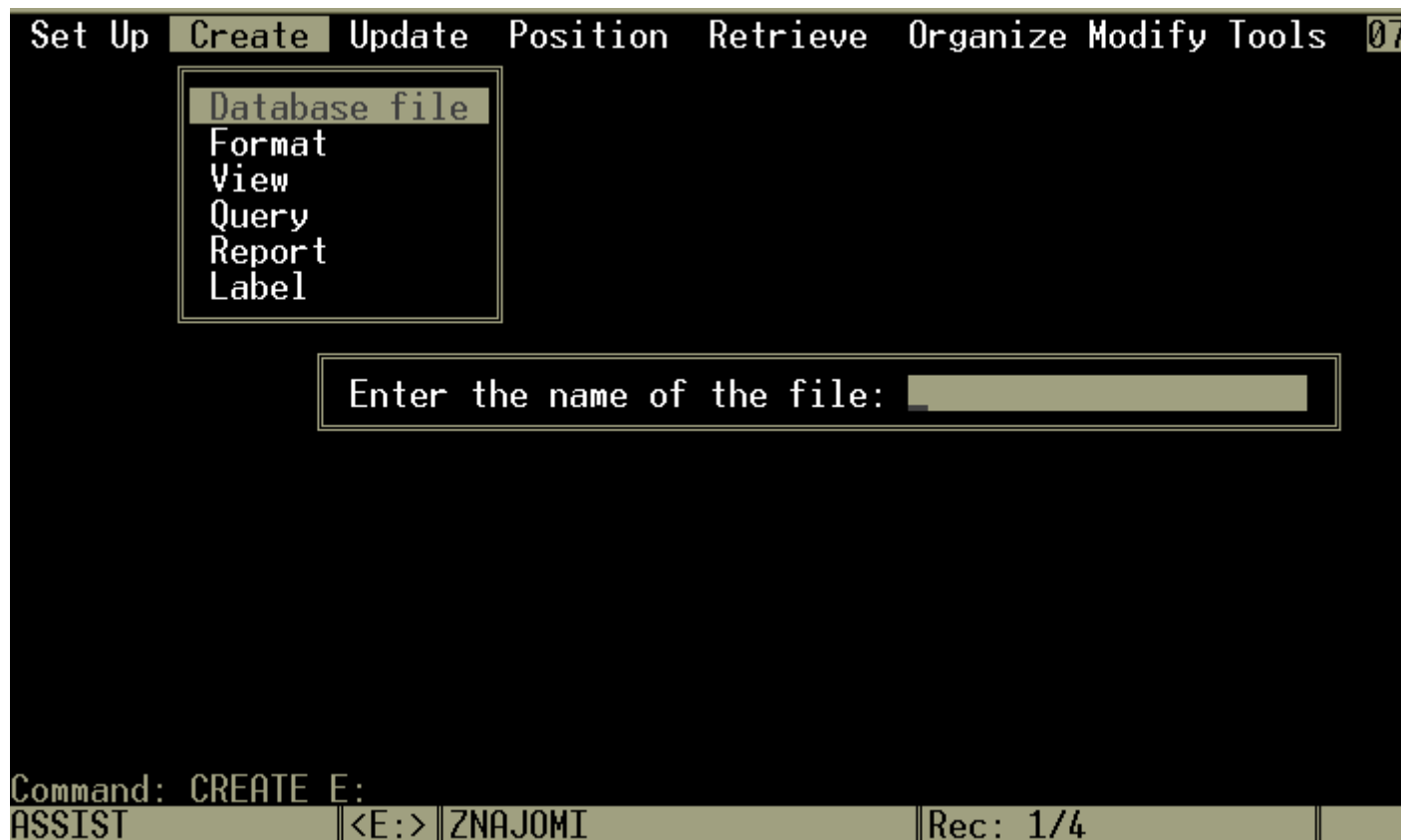
Program pomocniczy standardowo włączony jest po uruchomieniu dBase, można go także wywołać pisząc polecenie **assist** lub naciskając <F2>.

Tworzenie bazy danych

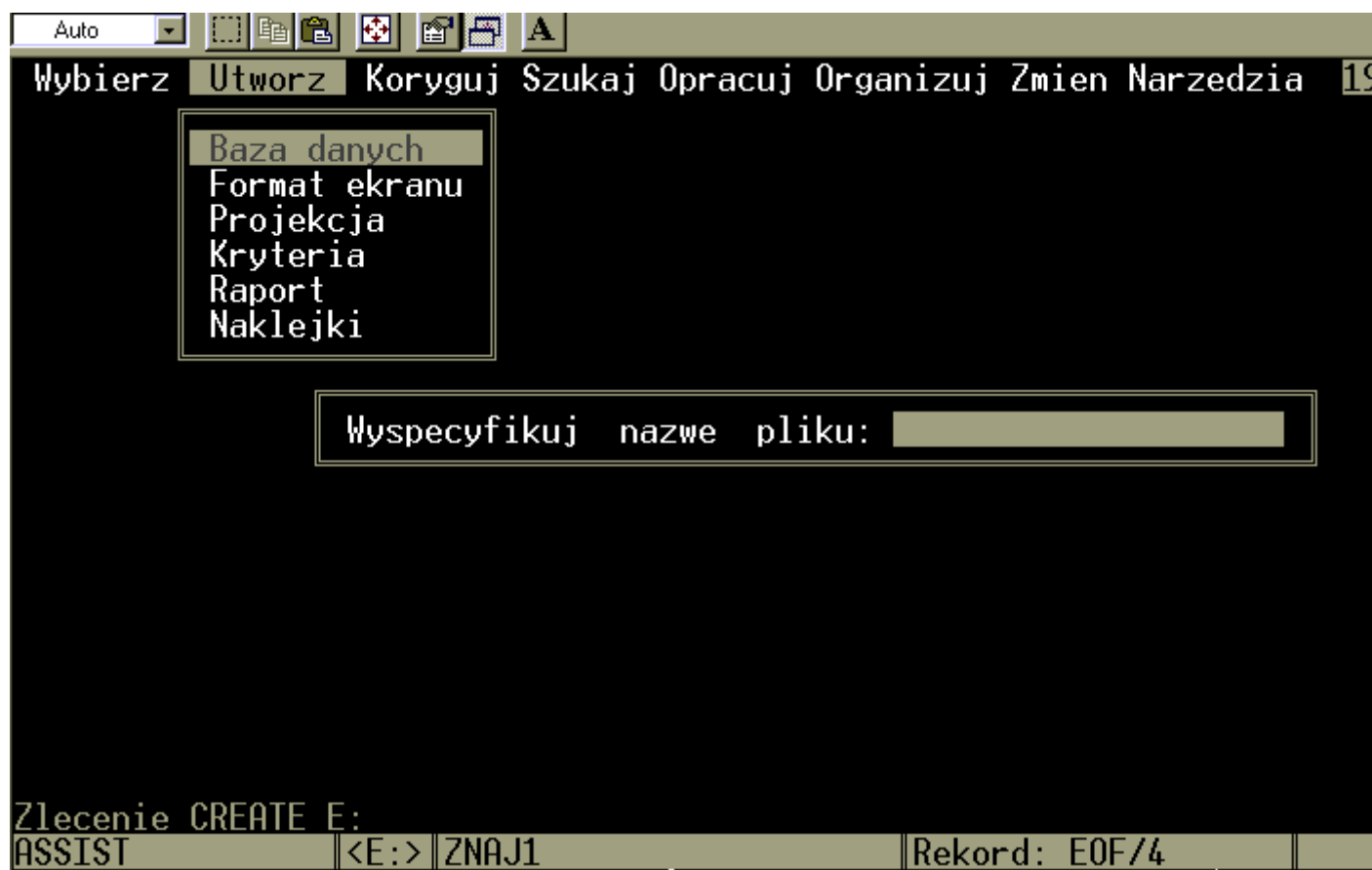
W systemie znakiem zachęty jest kropka. Można wykorzystac opcje menu lub wpisać polecenie po znaku zachęty.

- W dBase III+ będąc w Assist, wybiera się opcję **utwórz (create)**. Wybiera się plik bazy danych, dysk, wpisuje nazwe bazy i wchodzi do edytora ekranowego definicji bazy.
- W dBase IV po uruchomieniu Centrum Sterowania wybiera się z menu opcję **utwórz**.
- w trybie z kropką wpisać polecenie: create nazwa_bazy (można skracać do 4 liter - tu crea)

W wersji angielskiej:



W wersji polskiej:



Znajdziemy się w edytorze struktury, wpisujemy kolejno nazwy pól i ich typ oraz rozmiar. Typ pola (domyślnie znakowy) zmieniamy spacją. Wprowadzenie potwierdzamy przez <Enter>. Po wpisaniu wszystkich pól naciskamy <Ctrl><End>. Pojawi się pytanie: Czy wprowadzić teraz rekordy danych?(T/N) - tak jest w polskiej wersji dBase III+ Polonus, w angielskiej odpowiednia komenda po angielsku.

				Bytes remaining			
CURSOR <-- --> Char: ← → Word: Home End Pan: ^← ^→		INSERT Char: Ins Field: ^N Help: F1		DELETE Char: Del Word: ^Y Field: ^U		Up a field: ↑ Down a field: ↓ Exit/Save: ^End Abort: Esc	
Field Name	Type	Width	Dec	Field Name	Type		
1	NR	Numeric	3	0			
2	IMIE	Character	20				
3	NAZWISKO	Character	20				
4	DATA_UR	Date	8				
5	TELEFON	Character	8				
6	ADRES	Character	30				
CREATE		<E:>	ZNAJ1	Field: 6/6			

Wpisujemy kolejno dane kończąc pole naciśnięciem <Enter>. Wejście do pola memo przez <Ctrl><Home>, wyjście <Ctrl><End>. Po wprowadzeniu danych wszystkich rekordów naciskamy <Ctrl><End> i dane zostaną zapisane do bazy danych.

Przeglądanie bazy danych

Aby obejrzeć dane używamy polecenia **list** lub jeśli chcemy wypisać rekordy bez numerów to **list off**.

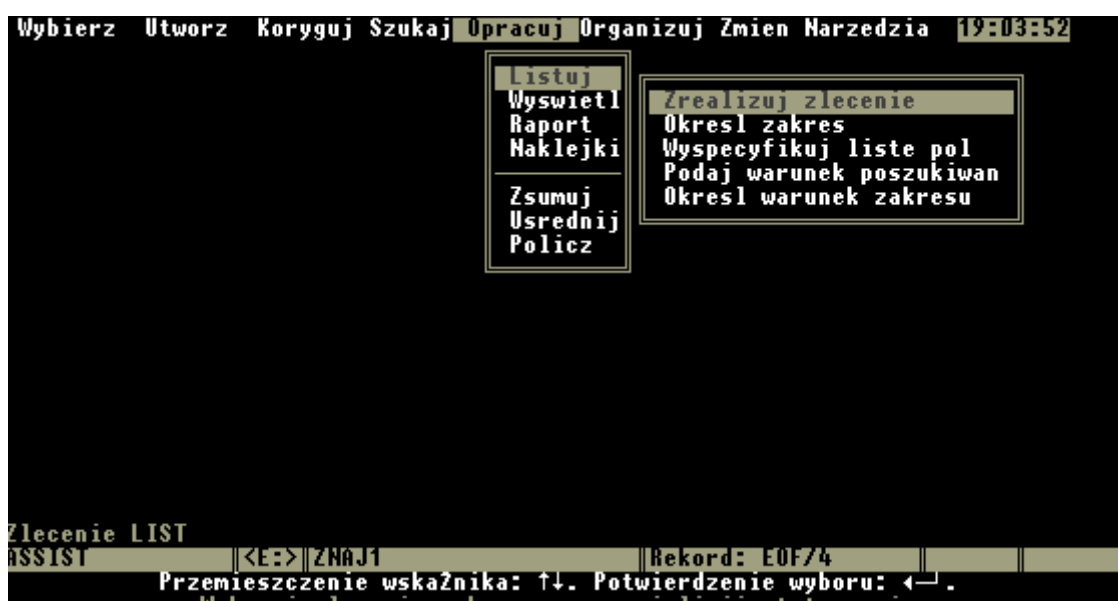
Można użyć też menu.

W wersji angielskiej dBase III+ :

Retrieve, List



w wersji Polonus: **Opracuj, Listuj**



Jeśli chcemy wydruk na drukarce to dodajemy opcję **to print**

Można wypisywać tylko wybrane pola z rekordów, np.
list nr, imie, nazwisko, telefon off

```
list nr, imie, nazwisko, telefon off
nr imie nazwisko telefon
1 Wiktor Kowalski 234-456
2 Andrzej Witkowski 123-678
3 Konrad Malicki 435-362
```

Gdy interesują nas pewne rekordy spełniające określony warunek, np. osoby o określonym imieniu to wydamy przykładowe polecenia:

`list for imie='Andrzej', nr, imie, nazwisko off`

`list for imie='Andrzej' .or. imie='Konrad' nr, imie, nazwisko off`

Można również określić zakres:

- **all** - cała baza
- **next x** - kolejnych x rekordów
- **rest** - wszystkie rekordy od bieżącego do końca

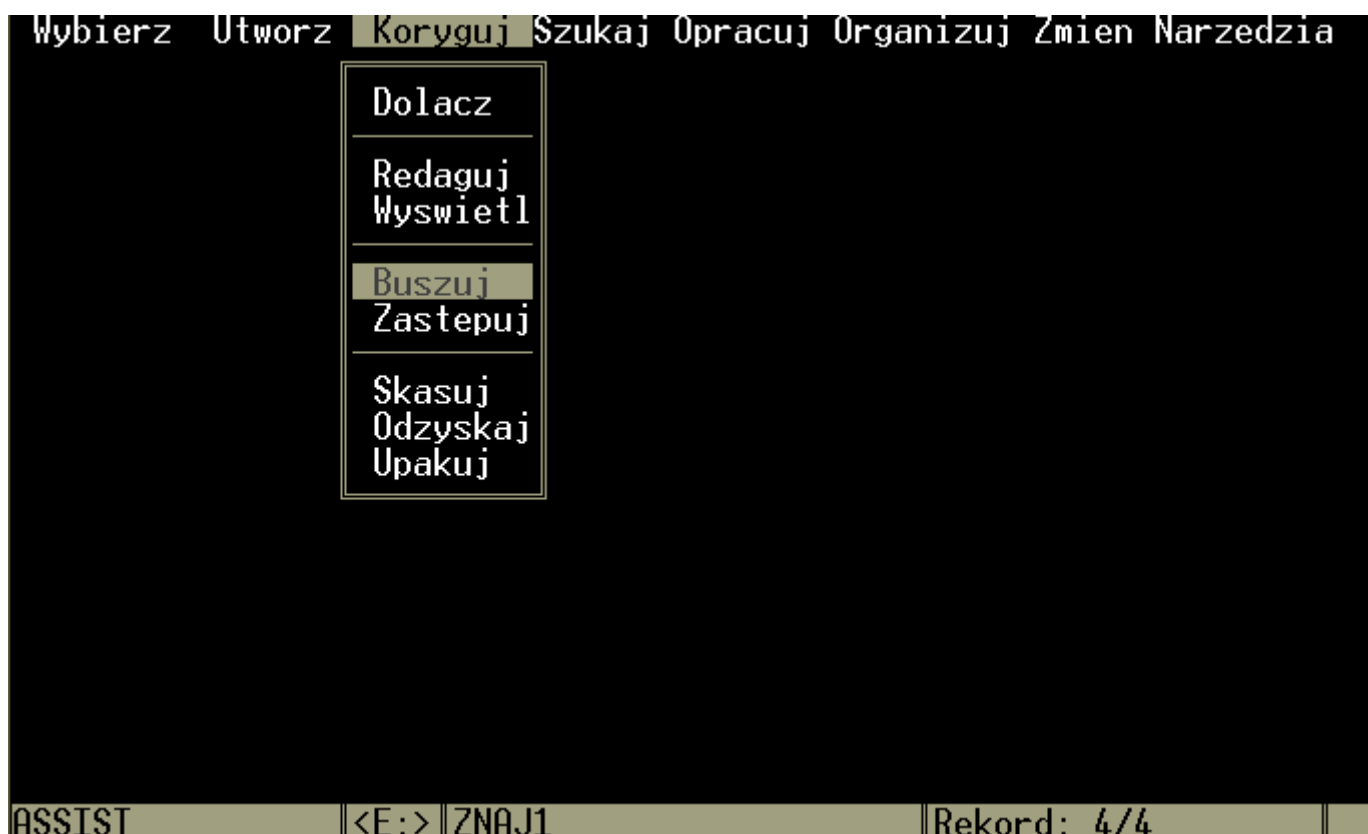
Podobne w działaniu jest polecenie **display**.

Przykłady:

- `goto top`
- `display rest` (jak `display all`)

`display telefon for nazwisko='Kowalski'`

Polecenie **browse** umożliwia przeglądanie i poprawianie rekordów. Pisze się komendę `browse` lub wybiera z menu Update Browse (albo Koryguj Buszuj w Polonus)



Przechodzenie między polami przy użyciu <Home> i <End> a między rekordami klawiszami kursor w górę i w dół.

W ramach pola kursor przesuwamy klawiszami strzałki kursora lewo, prawo.

Jeśli pole nie mieści się na ekranie to ekran można przesuwać <Ctrl>-> lub <Ctrl><-



Modyfikowanie bazy

- **Append** - dodawanie nowych rekordów (Koryguj Dołącz lub Update Append albo zwykłe polecenie)
- **Edit** - poprawa bieżącego rekordu
- **Browse** - umożliwia edycję grupy rekordów w trybie ekranowym. Edytuje zawartość bazy - przeglądanie i poprawianie rekordów
- **Insert** [before] [blank] - wstawia nowy rekord za lub (z Before) przed bieżącym rekordem, po czym przechodzi w tryb edycji ekranowej (jeśli nie było opcji Blank).
- **Change** - umożliwia bezpośrednią modyfikację w trybie ekranowym zawartosci wyspecyfikowanych rekordów i pól w aktualnej bazie danych

Często trzeba zmienić rekord bieżący. Stosuje się polecenie **locate** z parametrem **for locate** [<zakres.>] [for <warunek>] [while <warunek>]
 np. `locate for nazwisko="Malicki"`
 edit

Zmiana rekordu bieżącego jest możliwa poleceniem **goto**
goto/go [record] <wyrażenie> [bottom/top]
 Np.
`goto 10`
`go top`
`goto bottom`

Delete - zaznacza bieżący rekord do skasowania

Pack usuwa fizycznie z bazy danych rekordy zaznaczone do skasowania

Porządkowanie bazy

W celu łatwiejszego odnajdywania rekordów należy je uporządkować. Np. ułożyć w kolejności alfabetycznej wg nazwiska i imienia. Najlepszą metodą jest indeksowanie bazy wg określonych pól. Indeksowanie nie zmienia fizycznego ułożenia rekordów, ale tworzy specjalny zbiór indeksowy zawierający informacje o logicznej kolejności rekordów. Zbiór indeksowy jest automatycznie modyfikowany (porządkowany) przy wpisywaniu nowego rekordu (jeśli bazę otworzy się łącznie z odpowiednim plikiem indeksowym).

Plik indeksowy tworzy się poleceniem

index on <wyrażenie> to <plik indeksowy> [unique]
 - tworzy plik indeksowy skojarzony z aktywną bazą danych. Pola kluczowe są układane alfabetycznie, chronologicznie lub numerycznie (rosnąco). Opcja Unique powoduje pomijanie powtarzających się kluczy
 index on nazwa_pola_i[+nazwa_pola_j+..] to nazwa
 np.
`index on nazwisko+imie to nazw`

Aby możliwa była automatyczna zmiana indeksów, bazę należy otwierać poleceniem:

`use znajomi index nazw`

Wykonanie polecenia:

`list nr, imie, nazwisko, telefon off`

po indeksowaniu bazy wg pola nazwisko i imie spowoduje wyświetlenie listy posortowanej wg nazwisk i imion w ramach wspólnych nazwisk.

Inną metodą porządkowania jest **sortowanie** bazy. Jest to czynność pracochłonna polegająca na fizycznej zmianie kolejności rekordów w pliku bazy danych.

Każde wprowadzenie nowego rekordu powoduje konieczność ponownego posortowania bazy, z tego względu metoda ta jest mniej popularna i może być stosowana w odniesieniu do baz o małej liczbie rekordów.

Do sortowania bazy danych stosuje się polecenie:

sort to nowy_plik on nazwa_pola

Można sortować w porządku rosnącym dodając parametr /A (domyślnie) lub malejącym (/D).

Pełna forma polecenia:

sort to <nowy plik> on <pole1> [/a] [/c] [/d] [, <pole2> [/a] [/c] [/d] ... [<zakres>] [for <warunek.>] [while <warunek>]

Tworzenie nalepek

Do tworzenia nalepek stosuje się polecenie

create label plik_z_proj_nalepek

np. `create label znaj`, gdzie `znaj` jest nazwą pliku z projektem nalepki.

Wygląd ekranu edytora nalepek, który ukaże się po wydaniu komendy

w wersji angielskiej

```

Options                               Contents                               Exit 09:00:13 pm
-----
Predefined size: 3 1/2 x 15/16 by 1
Label width:    35
Label height:   5
Left margin:    0
Lines between labels: 1
Spaces between labels: 0
Labels across page: 1

CURSOR:  <-- -->   Delete char: Del   Insert row: ^N   Insert:  Ins
Char:    + +       Delete word: ^T    Toggle menu: F1  Zoom in: ^PgDn
Word:    Home End  Delete row: ^U    Abandon:   Esc   Zoom out: ^PgUp

CREATE LABEL  <E> ZNAJ.LBL  Opt: 1/7
Position selection bar - ↑↓. Select - ←. Leave menu - ↔.
Select a standard label size: (Width x Height by Number across).

```

Polonus

```

Opcje                               Zawartosc                               Koniec 21:03:33
-----
Rozmiar domniemany: 3 1/2 x 15/16 na 1
Szerokosc:      35
Wysokosc:       5
Lewy margines:  0
Linii miedzy nalepkami: 1
Spacji miedzy nalepkami 0
Naklejek w linii: 1

KURSOR:  <-- -->   UsunĀZ znak: Del   Wstawiz liniĀ: ^N   Wstawianie: Ins
Znak:    + +       UsunĀZ sĀowo: ^T    WylaczyĀ menu: F1   RozwinĀZ: ^PgDn
SĀowo:   Home End  UsunĀZ liniĀ: ^U    AnulowaĀ:   Esc   ZwinĀZ: ^PgUp

CREATE LABEL  <E> ZNAJ.LBL  Opcja: 1/7
Przesuwanie wskaźnika: ↑↓. Potwierdzenie wyboru: ←. Opuszczenie menu: ↔.
Okresl standardowy rozmiar naklejki: szerokosc x wysokosc i liczbe kolumn.

```

Należy określić rozmiary nalepki, liczby nalepek, odstępy. Gdy chcemy w wierszu umieścić kilka pól łączymy je znakiem "+", np. IMIE+NAZWISKO.

Wydrukowanie nalepek

label form plik_nalepek, np. label form znaj

Jeśli chcemy wydrukować nalepki tylko dla niektórych rekordów, np. osób o imieniu "Anna" to wydamy polecenie:

label form znaj for imie="Anna"

Programowanie baz danych w dBase

System dBase umożliwia pisanie programów na obsługę bazy.

Przykład programu, który na podstawie wprowadzonego imienia i nazwiska sprawdza czy osoba w bazie istnieje i jeżeli nie to dopisuje nowy rekord automatycznie wypełniając pola imie i nazwisko. W przeciwnym wypadku umożliwia dokonanie zmian w danych istniejącej osoby.

W przykładzie tworzone są 2 zmienne mimie i mnazwisko, w których przy użyciu poleceń get i read zapamiętane są dane wprowadzone przez użytkownika. Do przeszukiwania wykorzystano funkcję found().

Program można napisać pod dowolnym edytorem ASCII a można wykorzystać edytor tekstowy bazy poleceniem:

modify command plik_programu

np. **modify command dopisz.**

Po wpisaniu naciśnięcie Ctrl End zapisuje program na dysku, dodając rozszerzenie prg, tutaj dopisz.prg.

```
set talk off
set bell off
use znajomi
index on nazwisko+imie to nazw
use
use znajomi index nazw
clear
mnazwisko=space(20)
mimie=space(20)
@2,4 say 'Podaj nazwisko'
@2,30 get mnazwisko
@4,4 say 'Podaj imie'
@4,30 get mimie
read
seek mnazwisko+mimie
if .not. found()
append blank
replace nazwisko with mnazwisko
replace imie with mimie
else
edit
endif
set talk on
set bell on
```

Język bazy danych dBase oferuje m.in. takie konstrukcje programowe jak:

Pętle:

```
DO WHILE warunek_logiczny
polecenia
ENDDO
```

Podjęcie decyzji

```
IF war_logiczny
polecenia
[ELSE]
polecenia
ENDIF
```

Wybór jednego z wielu

```
DO CASE
CASE war_log
polecenia
CASE war_log
polec
..
[OTHERWISE]
polecenia
ENDCASE
```

Uruchomienie programu następuje w linii poleceń:

do plik.prg

np.

do dopisz

Inne polecenia dBase

skip [wyrażenie] - przemieszcza się w podanej bazie o podaną liczbę rekordów, np. **skip -2**,
skip 3

recall - zlikwidowanie zaznaczenia do skasowania, **recall all** - wszystkie rekordy

ctod - zamienia napis na datę

pack - usuwa rekordy zaznaczone do skasowania

zap - usuwa wszystkie rekordy, zostawia strukturę bazy

use - zamyka bazę danych z wszystkimi indeksami

Operatory relacji i logiczne

=	równy
<>	nierówny
>	większy

>=	nie mniejszy (większy, równy)
<	mniejszy
<=	nie większy
.not.	nie
.and.	i - koniunkcja
.or.	lub - alternatywa

Przykłady

```
delete all for NAZWISKO='K'
```

```
delete all for nazwisko='K' .or. nazwisko='k'
```

```
delete all for srednia_oc>=4
```

```
delete all for data_ur>=ctod('06/01/79')
```

Wykorzystanie funkcji języka dBase

W języku dBase operacje wykonuje się na danych z pól oraz na zmiennych pamięciowych.

Przykłady

```
x=5
y='5'
z=3*x
?x
?y
?z
```

```
a='Funkcje'
b='baz danych'
c=' '
d=a+c+b
?d - konkatencja łańcuchów
```

Operatory arytmetyczne od najwyższego priorytetu do najmniejszego

- () - nawiasy
- ^ lub ** - potęgowanie
- * i / - mnożenie i dzielenie
- + i - - dodawanie i odejmowanie

Wybrane funkcje

? abs(-5)
? int(2.34)
? mod(7,5) - reszta z dzielenia
? sqrt(16) - pierwiastek
? chr(65) - wynik A - zamiana kodu znaku na znak
? asc('A') - znak na nr kodu
? val ('123') - tekst na liczbę
? str(25.345,2) - liczba na tekst
? ctod('05/03/95') - tekst na datę
? date() - data
d=date()
? dtoc(d)

Funkcje do napisów

tekst='Ucze sie dBase'
? left(tekst,6)
? right(tekst,5)
? lower('NAUKA')
? upper(tekst)
? len('Ala ma kota')
? ltrim (' utnij spacje')
? replicate('*',40)

p=space(15)

Daty

d=date()
d1=d+10
d2=d-400
? cdow(d)
? cmonth(d)
? day(d2)
? month(d2)
? year(d2)
? dow(d)
? time()

Klawisze funkcyjne dBase

F1	help	pomoc
F2	assist	program asystenta
F3	list	zawartość rekordów otwartej bazy
F4	dr	zbiory *.dbf
F5	display structure	struktura bazy

F6	display status	ustawienia opcji oraz klawisze funkcyjne
F7	display memory	zmienne zadeklarowane
F8	display	zawartość bieżącego rekordu
F9	append	dopisywanie rekordu
F10	edit	modyfikacja bieżącego rekordu

System pomocy - help

Help można uruchomić wpisując polecenie help lub naciskając klawisz funkcyjne F1. Na ekranie ukaże się odpowiednie menu. Można też uzyskać pomoc na temat dowolnego polecenia pisząc po jego nazwie /?

Przyśpieszenie pracy systemu dBase - Clipper

dBase interpretuje polecenia odczytywane z pliku (PRG).

Usprawnieniem jest opracowanie systemów umożliwiających tłumaczenie programów na kod maszynowy procesora - **kompilatorów**.

Najbardziej znanym programem tłumaczącym programy bazy jest **Clipper** Compiler firmy Nantucket.

Dodatkową korzyścią jest to, że nie trzeba posiadać systemu dBase.

Clipper zwiększa ilość plików otwartych do 250, liczba pól w rekordzie ze 128 do 1024, zmiennych pamięciowych z 256 do 64000. Można definiować własne funkcje
