

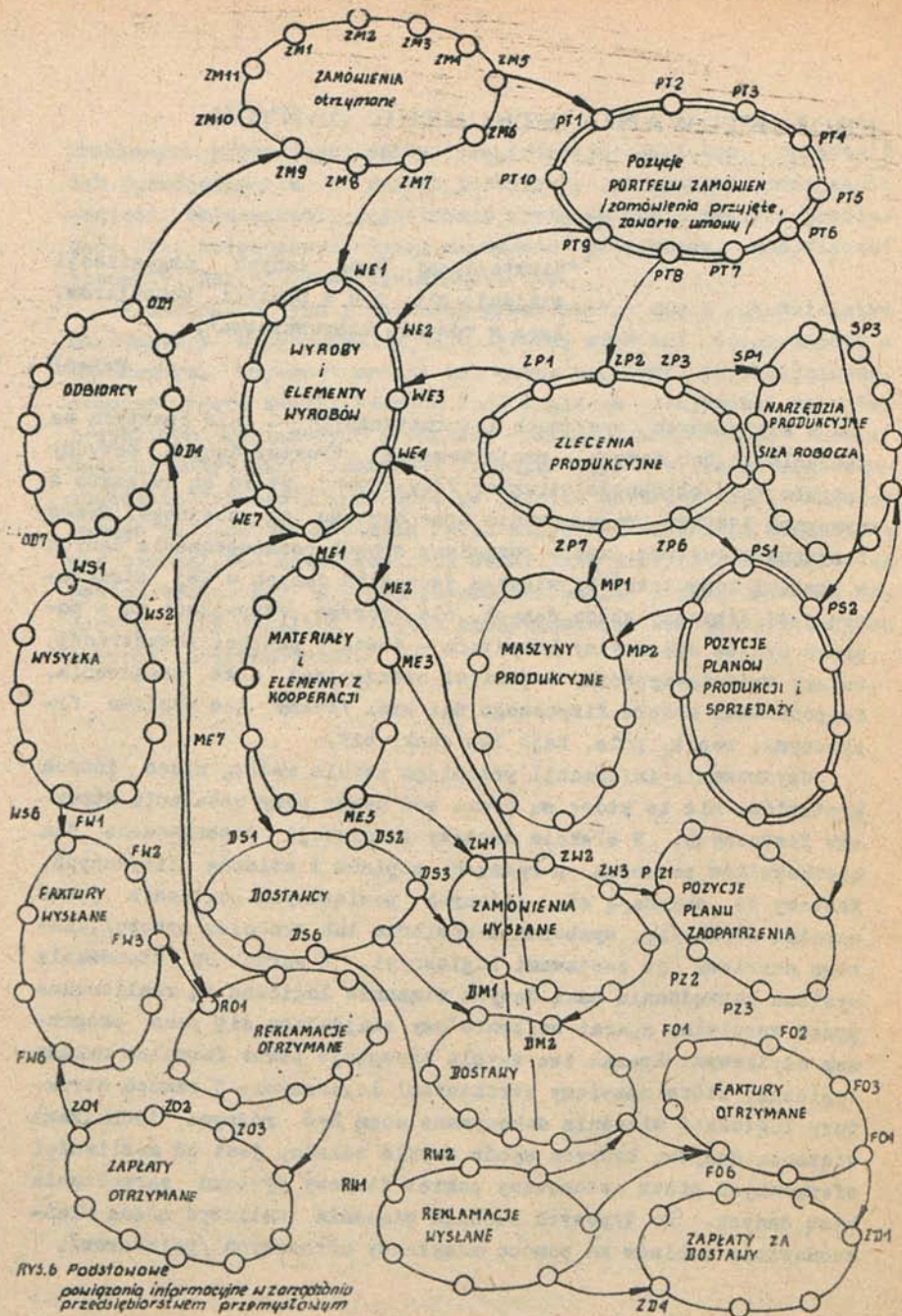
PROJEKTOWANIE STRUKTUR DANYCH

"Strategiczny bank danych organizacji znajduje się nie w pamięci komputerów, lecz w rozumie kierownictwa".

/L. Fried/

W klasycznych systemach informatycznych - nie opartych na technologii baz danych - projektowanie struktur danych dotyczy jedynie tych elementów struktur fizycznych, które są związane z wymogami języków programowania oraz systemu operacyjnego. Przez fizyczną strukturę danych rozumiemy sposób rozmieszczenia danych w pamięci komputera. Największą jednostką danych w tej strukturze jest fizyczny zbiór danych, dla którego wyznaczone są - poprzez system operacyjny komputera - obszary pamięci zewnętrznej, buforę wejścia-wyjścia w pamięci operacyjnej oraz urządzenia. Komponentami zbioru fizycznego są: blok /zwany też zapisem fizycznym/, zapis, pole, bajt lub znak, bit.

Użytkowanie informacji przebiega zwykle według nieco innych kryteriów, niż te które są brane pod uwagę przy ustalaniu struktur fizycznych. W efekcie zestawy informacji przeznaczone dla użytkowników pochodzą z różnych zapisów i zbiorów fizycznych. Zestawy te składają się z danych powiązanych logicznie jakąś wspólną cechą /np. symbolem zamówienia lub symbolem wyrobu/, dlatego nazwiemy je zestawami logicznymi. W warunkach stosowania systemu zarządzania bazą danych wiązania logiczne są realizowane przez specjalny aparat software'owy znajdujący się poza programem użytkowym. Aparat ten zwykle akceptuje pewne formalne układy logiczne, które nazwiemy strukturami logicznymi. W ramach struktury logicznej wiązania dokonywane mogą być różnymi technikami wiązania danych, których wybór zwykle zależny jest od możliwości oferowanych przez stosowany pakiet firmowy systemu zarządzania bazą danych. Do typowych technik wiązania zaliczyć można łańcuchowanie zapisów za pomocą odsyłaczy adresowych /pointerów/.



RYS.6 Podstawowe powiązania informacyjne w zarządnictwie przedsiębiorstwem przemysłowym

Jako typy struktur logicznych wymienia się z reguły struktury liniowe, hierarchiczne, sieciowe, relacyjne, itp. Bazy danych zwykle organizowane są według jednego typu struktury logicznej, dopuszczając czasem stosowanie różnych technik wiązań. W charakterze elementów zestawu logicznego występują przeważnie zapisy /rekordy/, rzadziej dane elementarne /pola/. Każdą zawartość zestawu logicznego^{1/} nazywa się wystąpieniem zestawu, natomiast każdą zawartość elementu zestawu nazywa się wystąpieniem elementu w ramach zestawu.

Istotę zestawu logicznego łatwiej będzie zrozumieć na podstawie przykładów. Jeśli w skład bazy danych, zbudowanej zgodnie z typem struktury sieciowej binarnej /pomiędzy dwoma elementami/ wchodzi zapisy wyrobów, materiałów, dostaw, dostawców, wysyłek, odbiorców, zamówień, to utworzyć z nich można wiele zestawów logicznych, takich jak np. materiały dostarczone /zestaw utworzony z zapisów "materiały" i "dostawy"/, wyroby własne /wiązane są zapisy "wyroby" i "wysyłki"/, zamówienia zrealizowane /zestaw składa się z zapisów "zamówienia" i "dostawy"/. W konwencji setu CODASYL'owskiego wystąpienie zestawu obejmuje jedno wystąpienie zapisu nadrzędnego /czyli ownera/ i wiele wystąpień zapisu podporządkowanego /czyli membra/. Przykład powiązań sieciowych w systemie informacyjnym przedsiębiorstwa przemysłowego podano na rysunku 6.

Jako kryteria wiązania zapisów występują identyfikatory obiektów /podmiotów i zasobów/ oraz atrybuty obiektów. Kwalifikacja identyfikatorów i atrybutów może być zmienna w ramach systemu, jednakże dla danego zestawu jest jednoznacznie określona. Przykładowo, dostawca występuje jako atrybut w zestawie "materiały dostarczone" oraz jako identyfikator w zestawie "zamówienia odnoszące się do danego dostawcy". Kolekcja wystąpień zestawów logicznych nazywa się bazą danych. Jest to więc maksymalna logiczna jednostka danych obsługiwana przez system zarządzania bazą danych, który zabezpiecza realizację powiązań pomiędzy danymi.

^{1/} W literaturze zestawy logiczne są nazywane też zbiorami logicznymi lub setami.

Fizyczne elementy strukturalne są tworzywem, z którego budowane są zestawy logiczne. Fizycznymi elementami tych zestawów są zwykle zapisy /rekordy/, czasem zaś segmenty, np. w DMS, lub wektory, np. w pakiecie ROBOT. Wśród segmentów wyróżnia się przeważnie segmenty główne /spełniające rolę nadrzędną/, segmenty danych /w liczbie stosownej do ilości występujących danych/ oraz opcjonalnie segmenty adresowe spełniające rolę tablic odsyłaczowych /wskazujących segmenty wchodzące do określonego zestawu logicznego/.

Jeśli w charakterze elementów fizycznych występują wektory, wymaga to użycia specyficznej technologii przetwarzania danych /gdyż nie ma tutaj konwencji zapisów/, lecz umożliwia elastyczne tworzenie struktur sieciowych oraz łatwiejszą rozbudowę baz danych. Wśród wektorów wyszczególnia się wektory kluczy, wektory powiązań oraz wektory danych. Wektor kluczy zakładany jest w zasadzie jednokrotnie dla danej bazy i zawiera identyfikatory obiektów. Wektor danych jest kolekcją wszystkich wartości /pól/ jednej danej, występuje więc tyle razy ile występuje nazw danych. Zespół wektorów danych traktowany być więc może jako poprzeczny przekrój zapisów całego zbioru. Wektor powiązań tworzony jest dla każdej relacji i reprezentuje sobą ciąg adresów /a ściślej, względnych pozycji/ tych kluczy, które odnoszą się do spełnionych relacji.

Szczegółowe omówienie struktur fizycznych wykracza poza ramy tej pracy, dlatego bardziej zainteresowanego Czytelnika odsyłamy do pozycji bibliograficznych /47/, /48/, /49/.

Projektowe rozwiązania logicznych struktur danych podlegają w praktyce znacznym ograniczeniom, wynikającym ze stosowanego pakietu banku danych^{1/}. Niekiedy dotyczy to typów struktur danych /np. brak możliwości stosowania struktur sieciowych/, czasem techniki wiązania lub metod dostępu do danych. W warunkach strukturalnego projektowania projektant, podejmując decyzję wyboru pakietu, powinien brać pod uwagę następujące czynniki:

^{1/}Przez bank danych rozumiemy tutaj kompleks eksploatacyjny, składający się z bazy danych i systemu zarządzania bazą danych.

- rodzaj elementu fizycznego podlegającego wiązaniom /np. zapisowa czy segmentowa organizacja bazy danych/,
- dopuszczalny typ struktury logicznej /liniowa, hierarchiczna, sieciowa, itp./ rozpatrywany w aspekcie spełnienia wymogów problemu,
- technika wiązania elementów, oceniana z punktu widzenia możliwości prowadzenia dodatkowych powiązań do istniejących baz danych.

Wymienione podstawowe logiczne struktury danych są w implementacjach rozmaicie rozwiązane, prowadząc do struktur mieszanych /np. hierarchizowanych struktur sieciowych, struktur liniowych z węzłami hierarchicznymi, itp./. Prawie wszystkie systemy zarządzania bazą danych opracowane w ostatnich latach ukierunkowane są na obsługę struktur sieciowych /co prawda za pomocą rozmaitych technik wiązania/.

Ze względów objętościowych i metodycznych nie możemy podać tutaj formalnych opisów struktur danych obowiązujących w poszczególnych pakietach, lecz ograniczymy się do pewnych uogólnień oraz opisu niezależnego od firmowych rozwiązań. Użyjemy do tego celu umownego języka specyfikacyjnego SDL /Structure Description Language/.

W języku tym typy logicznych struktur danych ująć można następująco:

LOGICAL-STRUCTURE IS	LINE	}	• struktura liniowa
	TREE		• struktura drzewiasta
	DBTG-NETWORK	}	• struktura sieciowa wg DBTG
	IMS-NETWORK		• str. sieciowa hierarchizowana
	IDS1-NETWORK /PLEX/		• str. sieciowa IDS1
	RELATIONAL-NETWORK		• struktura relacyjna
	VECTOR-NETWORK		• struktura wektorowa
	INVERTED	}	• struktura inwersyjna

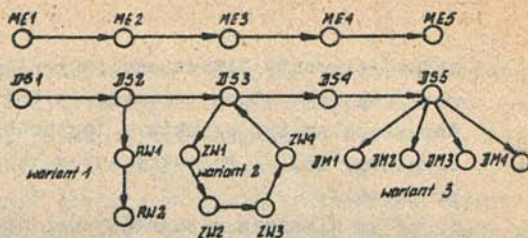
Podstawowe typy struktur logicznych przedstawia rysunek 7a i 7b. W strukturze liniowej każdy element jest powiązany logicznie z sąsiednimi elementami /poprzednim i następnym/, przy czym każdy element jest w zasadzie równoprawny. Piszemy - w zasa-

Struktury Linowe

- struktura Linowa prosta / lista prosta /

- struktura Linowa złożona hierarchizowana / struktury listujące warianty 1 i 2 /

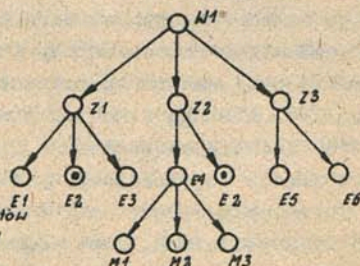
- struktura Linowa zamknięta / pierścieniowa /



Struktura hierarchiczna

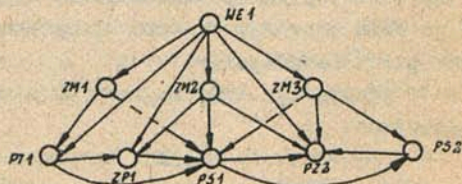
- W - wyroby
- Z - zespoły
- E - elementy (części)
- M - materiały

Uwaga: poprzez powtarzanie elementów wspólnych (np. E2) używamy struktury pseudosłabowej

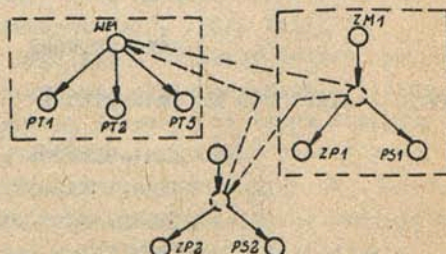


Struktura sieciowa

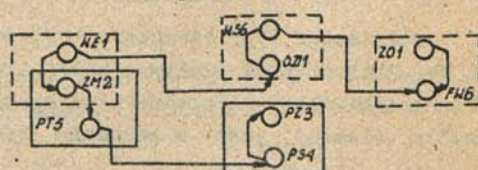
- Swobodna struktura sieciowa



- Sieć częściowych struktur hierarchicznych wg koncepcji IMS



- Sieć częściowych struktur wg koncepcji CODASYL



RYS. 7a Logiczne struktury danych

Struktura relacyjna
(tablice relacji)

Przykład tablicy relacji

Atakcja PLAN			
Symbol wyrobu	Ilość zamówiona	Ilość w planie produkcji	Ilość w planie sprzedaży
512562	100	130	110
512721	1250	1300	1250
.....

Struktura wektorowa

przykład

Lp.	Wektor klucza	Wektor danej 1	Wektor danej 2	Wektor powiązani
	Symbol wyrobu	Nazwa wyrobu	apoz	powiązania z dostawcami
1	512562	odkurzacz olfa	5
2
3
4
5
6

Struktura inwersyjna

zapisy zbiorów

● **Struktura inwersyjna**
(odwrócona zwykła)

Nazwa deskryptora	Wartość deskryptora	Liczba adresów	Adresy
.....
.....

● **Struktura odwrócona**
Lusijaca

Nazwa deskryptora	Wartość deskryptora	Adresy powiązań
.....
.....

Wyjaśnienie symboli elementów

- ME materiały i elementy z kooperacji
- ME wyroby i elementy wyrobów
- RU reklamacje wylane
- JM dostawy materiałów i elementów z kooperacji
- ZP zlecenia produkcyjne
- PS pozycje planów produkcji i sprzedaży
- WS nazwiska wyrobów i elementów wyrobów

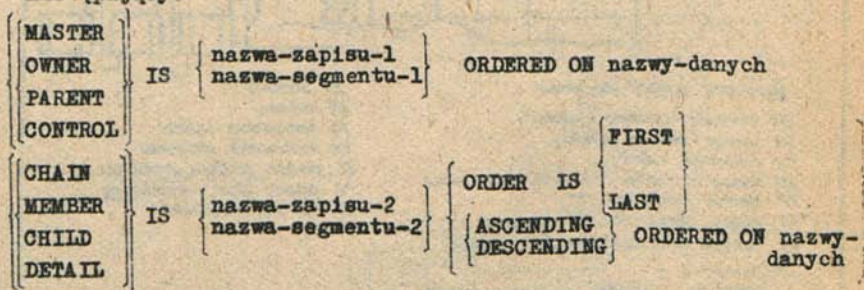
- JS dostawcy
- OD odbiorcy
- ZM zamówienia wylane
- ZM zamówienia otrzymane
- PT pozycje portfela zamówienia
- PZ pozycje planu zapotrzebowania
- ZO zapłaty otrzymane

dsie - gdyż w złożonych strukturach liniowych, powstających z połączenia dwóch szeregów liniowych, jeden z nich może być traktowany jako nadrzędny /np. dostawcy w stosunku do dostaw materiałowych/.

W strukturach drzewiastych /hierarchicznych/ elementy rozłożone są według grafu drzewa rozpoczynając od elementu początkowego, który nie jest podporządkowany żadnemu elementowi. W stosunku do każdego z pozostałych elementów występować może tylko jeden element nadrzędny oraz wiele elementów podporządkowanych. Odejście od tej zasady prowadzi do struktur sieciowych w których teoretycznie każdy element może być powiązany z każdym innym. W praktyce struktury sieciowe podlegają pewnym ograniczeniom poprzez wprowadzenie relacji binarnych lub hierarchizacji. Na przykład w IMS struktury sieciowe powstają poprzez łączenie cząstkowych struktur hierarchicznych.

W strukturach relacyjnych, zaproponowanych w latach 1970-1971 przez E.F. Codd'a, konwencjonalne struktury sieciowe zostały zastąpione relacjami tablicowymi, spełniającymi określone warunki tzw. normalizacji. Odejście od odsyłaczy adresowych i fizycznej struktury rekordowej jest charakterystyczne dla wektorowych sieci, zastosowanych w pakiecie ROBOT firmy SOFTWARE SCIENCES. Struktury inwersyjne umożliwiają tworzenie powiązań pomiędzy wieloma rekordami na poziomie atrybutów /pól nie będących kluczami/.

Opis zestawu logicznego danych zależy od typu struktury logicznej, do której przynależy. W języku SDL zestaw logiczny w ramach hierarchizowanej struktury sieciowej można opisać w sposób następujący:



LINKED	<div style="display: inline-block; vertical-align: middle;"> <div style="border-left: 1px solid black; border-right: 1px solid black; padding: 0 5px; margin: 0 5px;"> FORWARD BACKWARD F&B RING </div> </div>	UPON	<div style="display: inline-block; vertical-align: middle;"> <div style="border-left: 1px solid black; border-right: 1px solid black; padding: 0 5px; margin: 0 5px;"> nazwy-danych wyrażenie </div> </div>	.
--------	--	------	---	---

W podwójnych nawiasach podane są synonimy. W pierwszej frazie podaje się nazwę i uporządkowanie elementu nadrzędnego, w drugiej - podrzędnego, zaś w trzeciej - kryterium i kierunek relacji.

Techniki wiązań można wyróżnić następujące:

DATA-LINKING IS	CHAINING	• łańcuchowanie poprzez pointery wbudowane do rekordów
	POINTER-ARRAY	• tablice pointerów
	LINK-VECTOR	• wektor powiązań
	POINTER-FILE	• zbiór adresowy
	KEY-TABLE	• tablica kluczy
	INDEX-SET	• zbiór indeksów
	BIT-TABLE	• tablica wskaźników
	CROSS-FILE	• skorowidz adresowy dla potrzeb struktury inwersyjnej
	• inne mechanizmy łączenia danych /np. w modelu relacyjnym/

Najbardziej popularną techniką jest łańcuchowanie poprzez odsyłacze adresowe /tzw. pointery/ lokowane w zapisach lub w tablicach adresowych. Rozróżnia się łańcuchowanie proste jednokierunkowe /stosowane są wtedy odsyłacze "do przodu" albo "do tyłu"/ i dwukierunkowe /do następnika i do poprzednika/ oraz pierścieniowe, czyli takie, w których ostatni element grupy wiąże się z jej pierwszym elementem. Oprócz tego wyróżnia się odsyłacze adresowe z każdego elementu podporządkowanego do elementu nadrzędnego. Jeśli łańcuchowanie stosowane jest do wiązania zapisów na poziomie identyfikatorów obiektów /kluczy pierwotnych/, to w stosunku do atrybutów /kluczy wtórnych/ wykorzystuje się zwykle technikę inwersyjną, której istota polega na dostarczaniu odpowiedzi na pytanie: "które obiekty zawierają dane atrybuty?" za pomocą tzw. zbiorów odwróconych.

W zależności od przyjętej techniki wiązania i typu struktury logicznej w charakterze więzi adresowych używane być mogą:

- klucze pierwotne,
- fizyczne adresy dyskowe /zwykle ściśnięte/,
- logiczne adresy bazy danych /np.nr obszaru, strony, linii na stronie/,
- liczba porządkowa /adres względny/ elementu wektora,
- wskaźniki bitowe.

Pierwsze trzy rodzaje adresów używane są w łańcuchowaniu i technice inwersyjnej. Główna zaleta łańcuchowania polega na tym, że daje ono możliwość jednorazowego zapamiętania zapisu bez względu na to, w ilu zestawach logicznych występuje i w jakim uporządkowaniu. Wadą jest duża maszynochłonność przetwarzania przy dużej aktywności zbioru oraz nakłady pamięciowe na przechowywanie odsyłaczy adresowych. Relacyjne bazy danych są bardzo oszczędne w kontekście strat pamięciowych na więzi adresowe, lecz z kolei występuje tutaj większa redundancja danych. Najprostszą metodą wyrażania powiązań pomiędzy elementami są tablice wskaźników bitowych, w których wartość bitu "1" wyraża występowanie powiązania, zaś "0" oznacza brak powiązania. Wprowadzić nie potrzebują tak dużo pamięci jak zbiory inwersyjne i łańcuchowane, lecz zawierają mało informacji dla systemu zarządzania bazą danych /brak adresów i kierunku powiązań/.

Podsumowując ten skrótowy przegląd struktur danych, stwierdzić trzeba, iż odczuwa się w warunkach projektowania strukturalnego brak wymiernych kryteriów oceny sprawności poszczególnych typów struktur i technik wiązania. W celu zabezpieczenia efektywnej realizacji różnorodnych /niekiedy sprzecznych/ wymogów użytkowych, należy stosować rozwiązania mieszane, dostosowane do aktualnego zastosowania^{1/}. Na przykład, w procesach aktu-

^{1/} Mając do dyspozycji system zarządzania bazą danych nie można lekceważyć prostych struktur fizycznych. Przykładowo, w przetwarzaniu masowych danych niejednokrotnie /przy wysokiej aktywności zbiorów/ lepiej zdają egzamin zwykle zbiory sekwencyjne w porównaniu ze zbiorami łańcuchowanymi. Ponadto ważną rolę w budowie narzędzi programistycznych /systemów operacyjnych, translatorów/ spełniają jednostronnie otwarte fizyczne struktury liniowe /kolejki, stosy/.

alizacji można posługiwać się techniką łańcuchowania, zaś w procesach wyszukiwania techniką inwersyjną. Niestety, większość stosowanych pakietów baz danych nie daje możliwości doboru elementów składowych i uzyskanie racjonalnego kompleksu środków jest utrudnione. Jedynie w warunkach potraktowania ich jako tworzywa, obsługiwanego przez oprogramowanie prasytemu /m.i. konwertory systemowe języków i baz danych/, mogą one stać się jedną z podstaw projektowania strukturalnego.