

PODATKOVNE ZBIRKE

Tatjana WELZER
TEHNIŠKA FAKULTETA MARIBOR

UDK: 681.3.016

POVZETEK - Referat opisuje različne strukture podatkovnih zbirk in njihovo oblikovanje. Opisana je drevesna, mrežna in relacijska struktura zbirke ter metoda za njeno oblikovanje.

DATA BASES: This paper describes different structures of data bases and data bases design. Tree, network and relation data bases and a method for data base design are described.

1. UVOD

Širjenje distribuiranih računalniških sistemov in njihovo povezovanje v lokalne in javne računalniške mreže, omogoča novo in boljše organiziranost splošnih informacijskih sistemov. Zgradba uspešnih računalniško podprtih sistemov temelji na organiziranih podatkovnih zbirkah. Podatkovne zbirke so lahko centralizirane ali distribuirane. V centralizirani podatkovni zbirki se vsi podatki (trenutne vrednosti) nahajajo na eni lokaciji. Distribuirana podatkovna zbirka pa predvideva podatkovno zbirko, ki se tekoče obnavlja. Razdeljena je na več delov z ločenim, vendar enakovrednim upravljanjem. Uporabnik se bo avtomatsko povezal s tisto enoto, kjer so zahtevani podatki.

2. PODATKOVNE ZBIRKE

Podatkovno zbirko definiramo kot neredundantno zbirko vzajemno povezanih podatkov, ki jih uporabljamo za izvajanje ene ali več aplikacij ali pa kot zbirko podatkov, ki so medsebojno povezani brez nepotrebne redundance in omogočajo optimalno izvajanje najrazličnejših aplikacij. Podatkovno zbirko lahko definiramo še na več načinov, bistvo vseh definicij pa je, da nam podatkovna zbirka predstavlja skladišče informacij, ki jih uporabljamo za izvajanje posameznih aplikacij.

V razvoju podatkovnih zbirk od njihovega nastanka do danes so se izoblikovale tri osnovne strukture: hierarhična ali drevesna, mrežna in relacijska struktura.

2.1. DREVESNA STRUKTURA

Drevo je zgrajeno hierarhično iz elementov imenovanih segmenti, med katerimi veljajo naslednji odnosi:

1. Najvišji nivo v hierarhiji ima le en segment, ki ga imenujemo koren.
2. Vsi segmenti razen korena so povezani z enim in to samo enim segmentom na višjem nivoju, gledano s stališča danega nivoja.

Segment na višjem nivoju imenujemo starši, segmenti na danem nivoju so otroci. Posamezen element lahko istočasno opravlja nalogo staršev in otrok.

Drevesna struktura je lahko uravnotežena ali neuravnotežena. Pri uravnoteženem drevesu ima vsak segment enako število vej. Razvoj drevesa pri tem poteka od vrha proti

dnu in od leve proti desni. Poseben primer uravnotežena drevesa je binarno drevo, pri katerem iz vsakega segmenta izhajata natanko dve veji ali nobena veja.

Ob binarnem drevesu poznamo še B-drevo. B-drevo je uravnoteženo, vendar organizirano v vejah končne dolžine, ki lahko vsebujejo le določeno število informacij. Ko vejo napolnimo z elementi, se preprosto razcepi v dve novi veji na istem nivoju.

2.2. STRUKTURA MREŽE

Če ima otrok več staršev in povezave ne moremo opisati s pomočjo drevesa, govorimo o mrežni strukturi. V njej je lahko vsak segment povezan z vsakim segmentom. Vsako mrežno strukturo je možno poenostaviti, tako da jo prevedemo v preprostejšo drevesno strukturo. To poenostavitev izvedemo s pomočjo redundance (v zapletenejših primerih se zato poenostavitvi odpovemo).

2.3. STRUKTURA RELACIJE

Pri klasični strukturi podatkovne zbirke (drevesna in mrežna struktura), najpogosteje naletimo na naslednje probleme: podatkovne strukture na logičnem nivoju so zapletene, ni podatkovne neodvisnosti v smeri fizično - logično, manjka preprost mehanizem za dostop do podatkov na logičnem nivoju, programer pa je prisiljen uporabljati vnaprej definirane fizične poti. Rešitev teh problemov predstavlja relacijska podatkovna zbirka. V njej so vsi podatki eksplicitno predstavljeni kot vrednosti v tabelah-relacijah, ki so osnova relacijske strukture. S tabelami zelo enostavno prikažemo relacije - povezave med posameznimi podatki.

PREDMETI	PŠTEV	PIME	PLET	PŠTEVILO
P1	OSNOVE ELEKTR.	1		90
P2	MATEMATIKA	3		60
P3	ELEKTRONIKA	2		75
P4	REGULACIJE	2		75
P5	FIZIKA	1		45

UČITELJI	UŠTEV	IME	NAZIV	KRAJ
	OŠ1	NOVAK	DOC	MARIBOR
	OŠ2	PETAN	R. PROF	LJUBLJANA
	OŠ3	KOREN	V. PRED	KRANJ

RAZPOREDITEV	UŠTEV	PŠTEV	DATUM	ŠT. TEDNOV
	OŠ1	P4	05.10.83	10
	OŠ2	P1	18.02.84	15
	OŠ3	P2	15.11.83	10
	OŠ3	P5	05.10.83	05

Vsaka tabela ima svoje ime. V našem primeru so to PREDMETI, UČITELJI in RAZPOREDITEV. Vsaka vrstica nam opisuje eno skupino podatkov. Posamezne lastnosti so opisane v stolpcih, katerih imena imenujemo atribute. Množico vseh vrednosti, ki nastopi v posameznem stolpcu pa imenujemo domena. V tabeli PREDMETI so atributi PŠTEV, PIME, PLET in PŠTEVILO UR. Ti atributi opisujejo za vsak predmet: njegovo številko oziroma oznako, ime, letnik v katerem se predmet predava in število ur. Če želimo ime stolpca ločiti od morebitnega istega imena v kakšni drugi tabeli, mu kot predpono dodamo ime tabele s piko (PREDMETI.PLET).

Podatku, ki nedvoumno določa vrstico v tabeli pravimo ključ. Lahko je enostaven, če je v enem stolpcu, oziroma sestavljen, če ga določa več stolpcev. Ključa PŠTEV v tabeli PREDMETI in UŠTEV v tabeli UČITELJI sta enostavna. V tabeli RAZPOREDITEV pa je ključ lahko sestavljen iz UŠTEV, PŠTEV in DATUM.

Iz opisanih primerov lahko spoznamo tudi bistvene lastnosti relacij:

V posamezni tabeli nimamo podvojenih vrstic, niti podvojenih stolpcev. Vrstni red vrstic in stolpcev ni pomemben. Med različnimi tabelami ni vidnih vezi, saj povezave med njimi dosežemo s primerjanjem vrednosti v posameznih stolpcih.

2.3.1. OPERACIJE V RELACIJSKIH MODELIH

Osnovne relacije so podane v tabelah, iz katerih lahko s primernimi relacijskimi operatorji gradimo nove relacije - tabele (relacija in tabela sta sinonima).

Pravila delovanja relacijskih operatorjev nam pojasnjuje relacijska algebra. Operatorji (unija, presek, razlika in kartezični produkt) so sposojeni iz teorije množic. Ob njih v relacijski algebri srečamo še druge operatorje med katerimi so najpomembnejši selekcija, projekcija in združitev.

Selekcija

S tem operatorjem dobimo iz tabele A tabelo B tako, da iz prve tabele prenesemo v drugo tabelo samo tiste vrstice, ki ustrezajo nekemu pogoju.

Projekcija

Operator projekcije prenese iz osnovne tabele samo določene stolpce. Pri tem se lahko zgodi, da dobimo v novi tabeli dve ali več enakih vrstic. Redundantne vrstice izločimo in s tem izpolnimo zahtevani pogoj za relacijsko strukturo.

Združitev

Pod tem imenom srečamo več različnih operatorjev: združitev po enakosti, naravno združitev in zunanjo naravno združitev.

Imejmo relaciji R in S, ki imata skupno domeno D. Z A in B označimo atributa v relacijah R in S in oba definiramo nad D. Združitev po enakosti relacije R na A z relacijo S na B je takšna podmnožica kartezičnega produkta $R \times S$, kjer ima vsaka vrstica iste vrednosti v stolpcih A in B.

O naravni združitvi govorimo, kadar obdržimo samo en stolpec. Ta se zgodi, če imamo po združitvi po enakosti dva stolpca z istimi vrednostmi. Zunanja naravna združitev pa nam omogoča, da v relacijo uvrstimo tudi tiste vrstice iz združenih tabel, ki nimajo ustreznega para v drugi tabeli. Opisane operatorje lahko med seboj tudi povežemo in dobimo povezane operatorje.

3. OBLIKOVANJE PODATKOVNIH ZBIRK

Z razvojem podatkovnih zbirk se je močno spremenilo tudi njihovo oblikovanje. Sprva je bil precejšen del oblikovanja povezan s fizično lokacijo pomnenja podatkov, z njihovim nalaganjem in dostopom do sekundarnega pomnilnika. Te naloge je danes v glavnem prevzel sistem za upravljanje podatkovne zbirke, tendenca oblikovanja podatkovnih zbirk pa je v optimizaciji logičnih modelov podatkovnih zbirk.

Pri oblikovanju podatkovne zbirke najprej analiziramo ogrodje (obstoječ sistem), nato pa postopoma prehajamo do posameznih nalog, podnalog in podatkovnih elementov, ki nas zanimajo. Nalogo opravimo postopoma v štirih fazah:

1. V analizi okolja zahtev zbiramo informacije, ki so potrebne za izvajanje naslednjih korakov. Osnovne tehnike za pridobivanje potrebnih informacij so prebiranje dokumentov in drugih pisanih virov, izpraševanje ljudi in analiziranje anket. To je istočasno tudi vhod v prvo fazo, njen izhod pa predstavlja uporabna specifikacija v obliki diagrama poteka, ki nam predstavi najpomembnejše aktivnosti sistema in njihove medsebojne povezave.
2. V drugi fazi analiziramo in specifikiramo sistem. Izhajamo iz obstoječega diagrama, ki nam zagotavlja dobro poznavanje ciljnega sistema. Vsako aktivnost diagrama hierarhično razdelimo v več posameznih in delnih nalog. Pri tem procesu pride tudi do delitve podatkov, ki pripadajo posamezni aktivnosti. Razdelimo jih v podatkovne elemente ali pa v podmnožice teh. Delitev aktivnosti in podatkov poteka tako dolgo, da imamo samo še enostavne in razumljive naloge ter podatkovne elemente. Končni rezultat je diagram poteka, ki je tokrat podrobneje razdelan.

Temu prvemu delu faze sledi še specifikacija nalog, podatkov (dokumentov) in uporabniških programov.

3. Cilj zasnove modela je prenos znanja, ki smo ga zbrali v 1. in 2. fazi, v določenem prikaz. Za uspešno modeliranje uporabljamo različne jezike (CSDL, Taxis, D-graphs, E-R model).

4. V zadnji fazi oblikujemo logično shemo. Izvesti moramo naslednje naloge:

Izdelamo natančni model podatkovne zbirke na osnovi koncepta modela, generiramo bazično logično shemo,

integriramo novo shemo, ki predstavlja model povezav in njihovo komunikacijsko težo, optimiziramo shemo in realiziramo logično shemo ob pomoči optimizacijske sheme.

Ko uspešno zaključimo zadnjo nalogo četrte faze, je podatkovna zbirka pripravljena za izvajanje operacijske faze.

4. ZAKLJUČEK

Pregled razvoja, struktur in oblikovanja podatkovnih zbirk je pokazal, da so na tem področju spremembe pogoste.

Tako so na področju podatkovnih zbirk trenutno izredno "popularne" relacijske zbirke, ki tudi nepoznavalcem organizacije podatkovne zbirke omogočajo aktivno vključevanje v oblikovanje le te.

Kot pomoč pri oblikovanju in delu z relacijskimi podatkovnimi zbirkami pa je nadvse pomembna tudi umetna inteli-

genca in njeni jeziki (LISP, PROLOG, POP, micro LISP, micro PROLOG...). Prolog lahko direktno uporabimo kot interpreter za relacijski model podatkovne zbirke. Še boljše pogoje za dela pa daje POPLOG- programirano okolje v katerem so združeni LISP, PROLOG in POP.

5. LITERATURA

1. S. Algić: Relacione baze podataka, Sarajevo 1984
2. D. N. Chorafas: Database for networks and mini-computers, New York 1982
3. J. Martin: Computer Data-Base Organization, New Jersey 1975
4. G. Wiederhold: Database Design, Tokyo 1981
5. G. Wiederhold: Knowledge and Database Management, IEEE Software januar 1984