

Modele de date utilizate în bazele de date pentru prelucrări grafice

Sef lucrari dr.ing. Marius Dorian ZAHARIA
Universitatea "POLITEHNICA" Bucuresti

Lucrarea prezinta principalele modele de date specifice sistemelor de gestiune a bazelor de date referindu-se în special la modelele adecvate bazelor de date spatiale si la modelul orientat spre obiecte.

Cuvinte cheie: model de date, baze de date, baze de date spatiale.

1. Introducere

Bazele de date utilizate în prelucrări grafice își găsesc aplicatii în domenii diverse astfel: Proiectarea și fabricatia asistate de calculator sau alte aplicatii de automatizare a proiectării (cum ar fi sistemele CASE folosite în dezvoltarea produselor software). Utilizatorii pot avea acces (eventual concurrent) la modele de obiecte 3D (în cazul proiectării mecanice) sau 2-1/2 D (în cazul proiectării asistate de calculator în domeniul electronicii). Aceste categorii de aplicatii trebuie să permită descrierea structurală a unui proiect, simularea functionalității lui, sinteza și gestiunea versiunilor proiectului.

Aplicatiile de birotica, publicare asistată de calculator (CAP) realizează gestiunea documentelor din diferite domenii de activitate, stocarea, manipularea și regasirea documentelor complexe, inclusiv a celor cu "comportare dinamică".

În domeniul graficii pe calculator există categorii numeroase de aplicatii front-end (client) destinate furnizării unor interfete grafice pentru diferite sisteme de gestiune a bazelor de date (back-end servers). Sistemele de vizualizare științifică constituie un caz important de astfel de aplicatii.

Din categoria bazelor de date pentru prelucrări grafice fac parte și sistemele de gestiune a informației geografice (GIS) care permit manipularea datelor spațiale; în acest domeniu au fost dezvoltate numeroase strategii de interogare sau structuri de date utile pentru memorarea/regasirea eficientă a datelor.

În domeniile chimiei și medicinei au fost dezvoltate aplicatii care permit manipularea și analizarea complexă de reprezentări la nivel fizic, biologic sau chimic ale organismelor vii, moleculelor organice etc.

În domeniul important al aplicatiilor multimedia există aplicatii cum ar fi sistemele pentru înregistrarea/redarea în timp real a informațiilor continuând imagini, sunet, video. Unele dintre aceste aplicatii au mare impact comercial; de exemplu: aplicatiile "video on demand" (selectarea unui film dintr-o bază de date de filme stocate numeric), video-conferința, prezentările multimedia pe CD-ROM, sau aplicatiile specifice, cum ar fi sisteme de arhivare de imagini ce contin fete sau amprente.

În sistemele de gestiune a bazelor de date folosite în aplicatii ca sisteme de informație geografică, sisteme de proiectare asistată de calculator, arhivarea sau manipularea imaginilor, sunt utilizate, alături de date alfanumerice (text), date de natură grafică constând în imagini fixe (sintetizate sau digitizate), date video codificate numeric (de obicei secvențe de cadre comprimate sau nu, însoțite de informație audio sincronizată cu cea video), date de natură geometrică (puncte, linii, dreptunghiuri, poligoane simple, suprafețe, volume) sau temporala referitoare la pozițiile succesive ocupate în spațiu (la momente diferite de timp) de obiectele componente ale bazei de date.

Caracteristicile fundamentale ale unui sistem de gestiune a bazelor de date (SGBD), inclusiv a celor utilizate în prelucrări grafice, sunt: *persistenta* (capacitatea entităților de a conserva informațiile caracteristice în

urma diferitelor apeluri ale sistemului de programe); *existenta tranzactiilor* (secvente de instructiuni care se executa, în totalitate sau deloc, în mod atomic); *controlul concurenței prelucrărilor*; *existenta unor sisteme de interogare*; *pastrarea integrității bazei de date*.

În plus fata de caracteristicile mentionate mai sus, sistemele de gestiune a bazelor de date utilizate în prelucrări grafice ar putea furniza urmatoarele facilitati:

- Suport pentru *tipuri de date spatiale* (care sa permita descrierea de relatii spatiale între portiuni ale obiectelor grafice) si *limbaje de interogare spatiale*;
- *Interogare interactiva* care sa permita specificarea cererilor la diferite niveluri de detaliu. De obicei cererile utilizator sunt exprimate prin intermediul unor interfete grafice care ofera posibilitatea de a selecta diverse obiecte grafice si a construi cererea în mod interactiv. Mecanismele de tip QBE (Query By Example) sunt folosite adeseori;
- *Extragere automata de caracteristici* ale obiectelor grafice si indexarea obiectelor în raport cu aceste caracteristici;
- Folosirea *indexurilor multidimensionali si spatiale* spre deosebire de S.G.B.D.-urile clasice în care structurile de indexare se bazeaza pe o singura cheie;
- *Gruparea obiectelor complexe*. Aplicatiile care realizeaza gestiunea unor obiecte de date complexe (documente hipermedia, obiecte C.A.D./C.A.M.) necesita implementarea eficienta a relatiilor de incluziune sau referire reciproca între aceste obiecte. Sunt folosite tehnici complexe de grupare a obiectelor în scopul minimizării timpului de acces (I/E) la un obiect sau a timpului necesar reconstruirii unui obiect pornind de la subobiectele care îl compun;
- Metode de *memorare eficienta a masivelor binare* (BLOB – Binary Large Objects). Astfel de S.G.B.D.-uri trebuie sa permita crearea si stocarea unor câmpuri binare lungi, sa foloseasca indexuri pozitionali care sa permita accesul rapid la un sir de biti, începând de la o pozitie data;

- Tehnici avansate de *optimizare a executiei cererilor*.

2. Modele de date

Un *model de date* reprezinta o abstractie matematica a datelor si a operatiilor care pot fi efectuate asupra lor. Diferite categorii de aplicatii pot lucra cu o mare varietate de formate de date; un model de date ar putea încerca sa dezvolte o reprezentare unificata a acestor formate sau ar putea permite coexistenta mai multor reprezentari ale datelor. Unele categorii de modele de date (interne) sunt destinate implementării de sistem iar altele (externe) sunt folosite de utilizatorii sistemelor de programe.

Modelele de date specifica modul de definire si organizare a multimilor de obiecte de date. Ele pot descrie nu numai structura fizica si implementarea obiectelor de date ci si structura logica a datelor (proprietatile logice ale valorilor de date primitive precum si modul prin care obiectele complexe sunt compuse din obiecte de date mai simple) sau comportarea datelor în procesul de calcul (structura interna a obiectelor de date este invizibila utilizatorului, fiind specificata numai comportarea functiilor care actioneaza asupra obiectelor de date).

La nivel *structural* un model de date se refera la:

- *tipurile de date primitive* care definesc o multime de valori primitive, operatii de baza, o relatie de ordine între aceste valori si o topologie peste multimea de valori;
- *modul de agregare* a valorilor primitive în obiecte de date compuse cum ar fi tuple de valori sau relatii functionale între variabile;
- *metadatele* descriu relatia între date si elementele pe care acestea le reprezinta. În cazul datelor ce reprezinta marimi fizice, metadatele ar putea desemna natura marimii, unitatile de masura, modul de realizare a esantionării specifice procesului de masurare, estimarea erorii de masurare, valori singulare etc.;

Modelul în retea, introdus de Data Base Task Group în 1969, pune la dispozitia utilizatorului date organizate ca tipuri înregistrare și dependente multivaloare între aceste tipuri. Modelul în retea permite unui tip înregistrare să fie implicat în mai multe relații.

Modelul de date ierarhic este asemănător celui în retea, dar permite unui tip înregistrare să fie implicat într-o singură relație (ca fiu al unui alt tip).

Ambele modele menționate anterior oferă unui utilizator posibilitatea navigării printre membrii unei relații, folosind construcții de forma: `get_first()`, `get_next()`. Relațiile proprietar-membru (în cazul modelului în retea) sau părinte-fiu (în cazul modelului ierarhic) sunt stocate implicit în înregistrările bazei de date ceea ce implică faptul că modelele nu asigură independența față de modul de implementare fizică a datelor. Aceasta abordare limitează extensibilitatea, ușurința în întreținere, gradul de re folosire a componentelor și portabilitatea aplicațiilor.

Modelul relational introdus de T. Codd (1970) este bazat pe teoria matematică a algebrelor relationale și a calculului cu predicate de ordinul unu. Algebrele relationale cuprind operații caracteristice multimilor (intersecție, reuniune, diferență, produs cartezian) și operații relationale (selecție, proiecție, joncțiune). Tipurile de date fundamentale sunt înregistrarea (tuplul) și tabloul (relația). Acest model de date are asociat și un limbaj de interogare relational. Un prototip de astfel de limbaj este SQL (Structured Query Language), care este un limbaj declarativ, spre deosebire de limbajele bazate pe navigare, specifice modelelor de date prezentate anterior. S.G.B.D.-urile fundamentale, care sunt bazate pe modele de tip relational sunt System/R (IBM) și INGRES (University of California at Berkeley). Alte companii producătoare de astfel de sisteme sunt Informix Corporation, Oracle Corporation, Microsoft, Inprise (fost Borland International), Gupta Corporation etc.

O altă categorie de modele de date (post-relationale) sunt *modelele semantice* care (similar modelelor orientate spre obiecte) încearcă să modeleze cât mai apropiat de realitatea înconjurătoare. Din această categorie face parte *modelul entitate-legatură* al lui Chen (1976). În termenii acestui model, o entitate este un obiect (lucru) care există și poate fi deosebit de alte obiecte (lucruri) similare. Din punct de vedere al construcțiilor standard din bazele de date, o entitate poate corespunde unei înregistrări iar atributele sale corespund câmpurilor înregistrării.

Modelele de date *semantice* sunt utilizate în primul rând ca mijloace de proiectare a bazelor de date. Modelarea entitate-legatură este utilizată și în analiză și proiectarea sistemelor orientate spre obiecte. De cele mai multe ori structura unei baze de date este proiectată utilizând modelul entitate-legatură și apoi este pusă în corespondență cu o schemă relatională utilizând limbajul de definire a datelor (DDL) al unui SGBD relational.

Modelele de date *functionale* încearcă includerea într-un model semantic a unor posibilități de a manipula datele utilizând "relații functionale". Astfel, în modelul DAP-LEX, atributele sunt tratate ca funcții iar valorile atributelor sunt determinate prin aplicarea funcțiilor corespunzătoare, unei anumite entități (sau mai multor astfel de entități). Unele sisteme de gestiune a bazelor de date orientate spre obiecte (SIM al firmei UNISYS) au la bază modele de date semantice și functionale.

Modelul de date *relational extins* menține baza teoretică solidă a modelului relational dar îi adaugă mai multă flexibilitate (semantica), permițând diferite extinderi cum ar fi includerea de tipuri de date abstracte ca și câmpuri ale unei relații, posibilitatea de a crea operații și proceduri noi cu ajutorul interfetei de programare (DPL) a bazei de date sau (în cazul modelului *obiect-relational*) posibilitatea de a include obiecte structurate, cu identitate, în câmpurile unei relații. Astfel de sisteme sunt POST-GRES

(Stonebraker, Rowe, Hiohama, 1980), Intelligent SQL (Khoshafian, 1991), UniSQL si Starburst.

Comitetul de standardizare ANSI, pentru limbaje folosite în cadrul sistemelor de gestiune a bazelor de date a propus în standardul SQL3 extensii, cum ar fi încorporarea în limbaj a unor tipuri de date abstracte si a conceptului de mostenire. Astfel de limbaje suporta si tipuri de date adecvate aplicatiilor de prelucrare grafica a informatiei, cum ar fi tipul BLOB ce desemneaza un masiv de biti care poate contine imagini, date video sau date audio.

În modelul relational, realitatea este modelata printr-o colectie de tabele, fiecare dintre ele contine un set de linii (tuple) iar câmpurile fiecărei linii (atributele tuplelor) pot contine numai instante ale unor tipuri de baza, atomice (integer, float, string). Aceasta reprezentare este cunoscuta sub numele de "prima forma normala". Modelele *relationale imbricate* (Schek, Scholl, 1986) permit utilizatorului sa descrie un atribut a carui valoare sa fie o relatie (multime de tuple) si deci sa reprezinte/manipuleze direct multimi întregi de tuple. Se creeaza astfel spatii de obiecte, având structura arborescenta, situate la un nivel superior fata de colectiile de tipuri atomice de baza.

Modelele bazate pe obiecte complexe permit nu numai imbricarea relatiilor ci si a unor obiecte arbitrare. Obiectele complexe pot fi construite prin utilizarea a doi constructori, "sets" si "tuples". Acestia permit crearea de spatii de obiecte de forma unui arbore. Exista modele care permit ca acelasi obiect sa fie un subobiect al mai multor "obiecte parinte", astfel de spatii de obiecte fiind structurate sub forma unui graf. Pentru a putea suporta aceasta structura a spatiului obiectelor, modelele orientate obiect trebuie sa permita conceptul de identitate a unui obiect (un obiect poate fi referit printr-un numar unic, generat intern, independent de valoarea cheii sale primare – care poate sa existe sau nu).

În functie de nivelul la care se face abstractizarea, modelele de date pot fi grupate în trei categorii: fizic, de implementare si conceptual.

Modelele la *nivel fizic* nu au, în general, suficienta flexibilitate pentru a putea cuprinde toate caracteristicile comune si operatiile aferente datelor din categorii cât mai largi de aplicatii grafice. Astfel de modele pot fi utile în standardizarea transferului de date între diferite aplicatii.

Categoriile cele mai des folosite de modele de date (la nivel de implementare) sunt modelele: relational, ierarhic, în retea si orientat spre obiecte. Descrierea unei baze de date la nivel de implementare alcatuieste asa numita schema de implementare a bazei de date. La nivelul implementarii unei aplicatii, modelele de date cele mai adecvate prelucrarilor grafice sunt *modelul orientat spre obiecte* si *modelul relational extins*. Astfel de modele pot fi dezvoltate pentru a corespunde unor categorii relativ specializate de aplicatii (de exemplu aplicatiile din domeniul vizualizarii stiintifice).

La nivel conceptual au fost propuse câteva categorii de modele de date independente de implementare, un exemplu ar fi modelul bazat pe teoria laticelor iar un altul este modelul "fasciculelor de fibre" folosit în aplicatiile de vizualizare stiintifica a câmpurilor vectoriale.

În aplicatiile de natura grafica un rol esential îl au *metadatele* care descriu informatia sintactica si semantica despre date, structura, atributele si istoricul manipularii acestora. Metadatele structurale descriu tipul si structura datelor (schema bazei de date) pentru a putea fi folosite de mecanismele de acces la date. O alta categorie de metadate se refera la relatiile dintre date si entitatile pe care acestea le reprezinta (de exemplu, numele câmpurilor de date si domeniile de variatie ale valorilor de date). Metadatele de documentatie pot reprezenta comentarii referitoare la date sau proceduri pentru descrierea unor prelucrari ce se vor executa la momente ulterioare de timp.

3. Modele de date în baze de date spatiale

Pentru a proiecta o baza de date spatia la trebuie avut în vedere modul în care sunt integrate reprezentarile datelor spatiale respectiv nespatale. Din acest punct de vedere pot exista sisteme dedicate, duale, stratificate, integrate.

Sistemele dedicate corespund unor aplicatii strict specializate si nu prezinta facilitati complete caracteristice bazelor de date. Sistemele nu sunt extensibile si nu pot realiza actiuni care nu au fost gândite anterior de catre proiectantii lor. Un exemplu tipic sunt sistemele de proiectare asistata de calculator în domeniul electronicii (RACAL-REDAC, Cadence, ELECTROCAD). Astfel de sisteme arhiveaza date geometrice necesare documentatiei de proiectare si executie a placilor cu circuite imprimate, permit amplasarea automata/manuala a componentelor si rutarea traseelor de interconexiune cu respectarea unor reguli de proiectare.

Arhitecturile de sistem duale disting datele spatiale de cele nespatale prin utilizarea unor modele de date distincte. O astfel de arhitectura implica existenta unor module diferite de gestiune a memoriei. Comunicatia între cele doua subsisteme (spatial/nespatial) se face cu ajutorul unor identificatori comuni. Aplicatiile realizate conform acestei arhitecturi ar putea avea dezavantaje legate de sincronizarea accesului la date sau de pastrarea integritatii informatiei geometrice. Astfel, actiunile efectuate asupra portiunii spatiale a datelor ar putea sa nu fie reflectate în portiunea nespatala. Un exemplu de S.G.B.D. spatial cu arhitectura duala este sistemul de informatie geografica ARC/INFO.

Existenta a doua module distincte de gestiune a memoriei ar putea fi evitata prin memorarea datelor spatiale conform unui model de date relational. În acest caz, datele spatiale trebuie transformate (de exemplu reprezentate prin puncte caracteristice) sau descompuse în parti constituinte (de exemplu interiorul unei regiuni este descompus în

mai multe blocuri sau în pixeli). Aceasta abordare sta la baza *arhitecturii stratificate*. Ierarhia de organizare a straturilor ar putea fi: nivelul de baza al SGBD-ului relational, nivelul spatial si nivelul aplicatiei propriuzise (care contine rutine specifice aplicatiei). Exemple de baze de date realizate conform acestei arhitecturi sunt SIRO-DBMS si GEO-VIEW.

Arhitecturile integrate permit utilizatorului sa extinda o baza de date cu propriile sale tipuri abstracte de date adecvate aplicatiilor spatiale. Pentru realizarea implementarii unei astfel de arhitecturi cele mai adecvate ar fi modelul de date relational extins sau cel orientat spre obiecte. Aceste sisteme permit adaugarea unor constructii ca: tipuri de date abstracte, obiecte complexe, câmpuri procedurale sau attribute multivaloare. Fiecare din tipurile de date (spatiale sau nu) trebuie reprezentate printr-o structura de date corespunzatoare care sa îndeplineasca cel mai bine cerintele problemei. Exemple de sisteme de gestiune a bazelor de date ce respecta o astfel de arhitectura sunt: Gral, POSTGRES, Probe, GEO++.

O problema importanta în sistemele baze pe o arhitectura integrata este modul de a realiza legaturile între descrierea datelor spatiale asociate unui obiect complex si restul informatiei nespatale a obiectului. O optiune este cea realizata în sistemul GEOQL unde fiecare relatie are asociat un singur atribut spatial. Sistemul acorda o importanta mai mare componente relationale decât celei spatiale; astfel, desi attributele spatiale ale relatiilor sunt implementate folosind structuri de date avansate (indexuri ierarhice multidimensionale), pentru a efectua operatii spatiale asupra a doua câmpuri (din doua relatii distincte) cu informatie spatia la va trebui construita o noua relatie (intermediara) ceea ce va micșora eficienta executiei cererilor spatiale.

În alte sisteme, datele spatiale si cele nespatale se refera reciproc prin legaturi bidirectionale. Fiecarui atribut spatial din schema bazei de date îi este asociata o structura de date spatia la utilizata pentru a

stoca toate instanțele aceluși atribut spațial, aferente unei mulțimi de obiecte. Această structură de date constituie un index pentru obiectele spațiale având un anumit tip (punct, linie, regiune) precum și un context asupra căruia se pot executa diferite operații spațiale (transformări geometrice, intersecții de poligoane, calcule de distanță etc.)

Ideea de a introduce noi tipuri de date abstracte ca domenii asociate unor atribute într-un S.G.B.D. relational și cea de a permite utilizatorului să-și definească propriile sale structuri de indexare se datorează lui Stonebraker. Conform acestei abordări, folosite în multe S.G.B.D.-uri extensibile, structurile de date spațiale sunt considerate structuri de indexare a informației.

Construirea unei baze de date spațiale ar putea avea la bază și un model orientat spre obiecte. Astfel se asigură un grad înalt de structurare a informației, prin introducerea conceptelor de clasă și moștenire. Independența reprezentării fizice și a celei logice a datelor și manipularea comodă a lor sunt facilitate prin încapsularea datelor și supraîncărcare. Tipurile de date spațiale pot fi implementate prin clase iar conceptul de moștenire este utilizat pentru definirea de subclase. Un exemplu de astfel de sistem a fost descris de Bancilhon (SGBD-ul orientat spre obiecte O₂). În acest sistem, datele spațiale sunt modelate pe două niveluri: geometric și "de mapare". O bază de date spațială este alcătuită dintr-o mulțime de mapări iar o mapare este o relație care are cel puțin un atribut spațial. Nivelul geometric corespunde atributelor spațiale care sunt reprezentate prin tipuri de date abstracte geometrice cum ar fi tipul punct, linie sau regiune.

Bibliografie

[Banc 91] F. Bancilhon, L. Delobel, P. Kanellakis, Building an Object-Oriented

Database System: The Story of O₂, Morgan Kaufman, San Mateo California, 1991.

[Catt 94] R. G. Cattell, Object Data Management, Addison Wesley Publishing Company, 1994.

[Chen 76], P. P. Chen, The entity-relationship model – Toward a unified view of data, ACM Transactions on Database Systems, No. 1, 1976.

[Egen 94] M. J. Egenhofer, Spatial SQL: A Query and Presentation Language, IEEE Transactions on Knowledge and Data Engineering, february 1994.

[Gallo 94] J. Gallop, Underlying Data Models and Structures for Visualization; aparut în: Scientific Visualization, Advances and Challenges, Academic Press, 1994.

[Gupta 91] A. Gupta, T. Weymouh, R. Jain, Semantic Queries with Pictures: The VIMSYS Model, Proceedings of the 17-th International Conference on Very Large Databases, 1991.

[Khos 96] S. Khoshafian, A. B. Baker, Multimedia and Imaging Databases, Morgan Kaufmann Publishers Inc., San Francisco, 1996.

[Mai 91] R. Maiocchi, B. Pernici, Temporal Data Management Systems: A Comparative View, IEEE Transactions on Knowledge and Data Engineering, december 1991.

[Oren 88] J. A. Orenstein, F. A. Manola, PROBE Spatial Data Modelling and Query Processing in an Image Database Application, IEEE Transactions on Software Engineering, may 1988.

[Samet 94] H. Samet, Spatial Data Models and Query Processing, aparut în Modern Database Systems: The Object Model, Interoperability and Beyond, W. Kim ed., Addison Wesley, Reading, MA, 1994.