

Modelarea datelor

Un **model** este o reprezentare a obiectelor și evenimentelor lumii reale și a asocierilor dintre ele. De fapt, el reprezintă o abstracție asupra aspectelor semnificative ale unei „întreprinderi“, ale unui sistem real, ignorând proprietățile accidentale. Modelul este cel pe care utilizatorii trebuie să-l cunoască; implementarea unui model este cea pe care utilizatorii nu este necesar să o cunoască. Diferența dintre model și implementare este, de fapt, un caz special și important al deosebirii uzuale dintre logic și fizic.

Modelele se impun prin sintaxa și prin semantica lor și, din acest punct de vedere, există trei tipuri fundamentale de modele:

- modele care descriu aspectele statice ale procesului modelat;
- modele care descriu aspectele dinamice ale procesului modelat;
- modele care descriu aspectele funcționale ale procesului modelat.

Un **model de date** reprezintă o colecție integrată de concepte necesare descrierii:

- datelor,
- relațiilor dintre ele,
- constrângerilor existente asupra datelor sistemului real analizat.

Modelarea unei baze de date permite trecerea de la percepția unor fapte din lumea reală la reprezentarea lor prin date. Modelul de date trebuie să reflecte fidel fenomene ale lumii reale, să urmărească evoluția acestei lumi și comunicarea dintre fenomenele lumii reale.

Modelul trebuie să asigure conceptele de bază care permit proiectantului bazei de date și utilizatorilor să comunice, fără ambiguități, cunoștințele lor privind funcționarea și organizarea modelului real analizat. Prin urmare, un model de date trebuie să reprezinte datele și să le facă înțelese.

În esență, modelul de date are trei componente:

- o mulțime de reguli conform cărora sunt construite bazele de date (partea structurală);
- o mulțime de operații permise asupra datelor, care sunt utilizate pentru reactualizarea sau regăsirea datelor (partea de prelucrare);
- o mulțime de reguli de integritate, care asigură coerența datelor.

Abordarea generală a problemei modelării semantice a datelor se face în patru etape.

- Se identifică o mulțime de concepte semantice care sunt utile în descrierea lumii reale. Se presupune că lumea reală (modelul real analizat) este formată din entități care au anumite proprietăți, că fiecare entitate are o identitate, că există legături, corelații între entități. Conceptul de corelație, ca și cel de entitate, este util, în mod intuitiv, la descrierea modelului.
- Se caută o mulțime de obiecte formale, simbolice care sunt utilizate

- pentru reprezentarea conceptelor semantice anterioare.
- Se dau reguli de integritate formale și generale (constrângeri) care să reflecte restricțiile la care este supus modelul.
 - Se definește o mulțime de operatori formali prin care pot fi prelucrate și analizate obiectele formale.

Modelul entitate-relație

Una dintre cele mai cunoscute abordări ale modelării semantice (cu siguranță una dintre cele mai utilizate) este cea bazată pe **modelul entitate-relație** (E/R). Acesta a fost introdus de către P.P. Chen în 1976 și rafinat de atunci în diverse moduri de către acesta și de mulți alți cercetători, ca un model de date conceptual, pentru a ușura proiectarea bazelor de date. Pentru reprezentarea grafică a modelului sunt utilizate diagramele E/R, care sunt modele neformalizate pentru reprezentarea unui model, unui sistem din lumea reală.

Diagramele E/R constituie o tehnică de reprezentare a structurii logice a bazei de date, într-o manieră grafică. Aceste diagrame oferă un mijloc simplu și inteligibil de comunicare a caracteristicilor importante ale designului unei anumite baze de date.

Diagrama E/R este un model de date conceptual de nivel înalt, independent de platforma *hardware* utilizată și de tipul SGBD-ului. Modelul este constituit din concepte care descriu structura bazei de date și tranzacțiile de regăsire sau reactualizare asociate.

Popularitatea modelului E/R ca modalitate de abordare a proiectării bazelor de date poate fi atribuită în principal tehnicii de realizare a diagramelor E/R. Această tehnică, ca și modelul E/R însuși, a evoluat de-a lungul timpului datorită noilor problematice care au apărut în proiectarea bazelor de date.

Baza de date poate fi definită ca o mulțime de date ce modelează un sistem real. Acest sistem este format din obiecte legate între ele. Modelul E/R împarte elementele unui sistem real în două categorii: entități și relații (legături, asocieri) între aceste entități. Entitățile și legăturile au anumite caracteristici, numite atribute. Nu trebuie confundat conceptul de relație, în sensul de asociere, care intervine în definirea diagramei E/R cu conceptul de relație care este specific modelului relațional.

!!! Studii de caz, exemple – prezentate la curs!

Deficiențe ale modelului E/R

În proiectarea unui model de date pot apărea diverse probleme datorită unei interpretări eronate a sensului unei relații. Aceste probleme sunt denumite **capcane de conectare**.

Unele dintre aceste capcane pot să nu fie semnificative pentru modelul particular considerat, în timp ce altele cer restructurarea modelului. Există două clase importante de capcane: de întrerupere și în evantai.

O capcană de întrerupere poate să apară acolo unde modelul sugerează existența unei relații între entități, dar nu există o cale între anumite apariții ale entităților. Această capcană poate să apară acolo unde există o relație cu participare parțială (0 la cardinalitatea minimă), care face parte din calea dintre entitățile ce sunt legate.

O capcană în evantai poate să apară acolo unde modelul ia în considerare o relație între entități, dar calea dintre anumite apariții ale entităților este ambiguă. Aceste capcane apar când două sau mai multe relații *one_to_many* provin din aceeași entitate.

Practic, aceste capcane generează situațiile în care, așa cum a fost proiectat modelul de date, el nu poate să răspundă la anumite interogări. De exemplu, pentru a afla pentru ce prezentare de modă a fost creată o anumită vestimentație, a fost necesară introducerea unei legături între entitățile PREZENTARE și VESTIMENTATIE, care însă a generat redundanță în modelul de date:

