BAZE DE DATE

CURS 8

Regulile lui Codd. Normalizare (partea 1).

Regulile lui Codd

Caracteristici ale modelului relaţional:

- nu există tupluri identice;
- ordinea liniilor şi a coloanelor este arbitrară;
- articolele unui domeniu sunt omogene;
- fiecare coloană defineşte un domeniu distinct şi nu se poate repeta în cadrul aceleiaşi relaţii;
- toate valorile unui domeniu corespunzătoare tuturor cazurilor nu mai pot fi descompuse în alte valori (sunt atomice).

Regulile lui Codd

Avantajele modelului relaţional:

- fundamentare matematică riguroasă;
- independenţă fizică a datelor;
- posibilitatea filtrărilor;
- existenţa unor structuri de date simple;
- realizarea unei redundanţe minime;
- supleţe în comunicarea cu utilizatorul neinformatician.

Regulile lui Codd

Limite ale modelului relaţional:

- rămâne totuşi redundanţă,
- ocupă spaţiu,
- apar fenomene de inconsistenţă,
- nu există mecanisme pentru tratarea optimă a cererilor recursive,
- nu lucrează cu obiecte complexe,
- nu există mijloace perfecţionate pentru exprimarea constrângerilor de integritate,
- nu realizează gestiunea totală a datelor distribuite,
- nu realizează gestiunea cunoştinţelor.

Regulile lui Codd

- În anul 1985, E.F. Codd a publicat un set de 13 reguli în raport cu care un sistem de gestiune a bazelor de date poate fi apreciat ca relaţional.
  - Niciun sistem de gestiune a bazelor de date nu respectă absolut toate regulile definite de Codd, dar acest lucru nu împiedică etichetarea acestor sisteme drept relaţionale.
- Nu trebuie apreciat un SGBD ca fiind relaţional sau nu, ci măsura în care acesta este relaţional, deci numărul regulilor lui Codd pe care le respectă.

Regulile lui Codd

- **Regula 1 –** regula gestionării datelor. *Un SGBD relaţional trebuie să fie capabil să gestioneze o bază de date numai prin posibilităţile sale relaţionale.*
- **Regula 2 –** regula **reprezentării informaţiei**. *Într-o bază de date relaţională, informaţia este reprezentată la nivel logic sub forma unor* ***tabele ce poartă numele de relaţii***.
- **Regula 3 –** regula **accesului garantat la date**. *Fiecare valoare dintr-o bază de date relaţională trebuie să poată fi accesată în mod logic printr-o* ***combinaţie formată din numele relaţiei, valoarea cheii primare şi numele atributului***.
- **Regula 4** – regula **reprezentării informaţiei necunoscute**. *Un sistem relaţional trebuie să permită utilizatorului definirea unui tip de date numit „****null****“ pentru reprezentarea unei informaţii necunoscute la momentul respectiv.*

Regulile lui Codd

- **Regula 5 –** regula **dicţionarelor de date**. *Asupra descrierii bazelor de date (informaţii relative la relaţii, vizualizări, indecşi etc.) trebuie să se poată aplica **aceleaşi operaţii** ca şi asupra datelor din baza de date*.
- **Regula 6 –** regula **limbajului de interogare**. *Trebuie să existe cel puţin un **limbaj pentru prelucrarea bazei de date***.
- **Regula 7 –** regula de **actualizare a vizualizării**. *Un SGBD trebuie să poată determina dacă o vizualizare poate fi actualizată şi să stocheze rezultatul interogării într-un dicţionar de tipul unui catalog de sistem*.
- **Regula 8** – regula limbajului de nivel înalt. *Regulile de prelucrare asupra unei relaţii luată ca întreg sunt valabile atât pentru operaţiile de regăsire a datelor, cât şi asupra operaţiilor de inserare, actualizare şi ştergere a datelor*.

Regulile lui Codd

- **Regula 9** – regula independenţei fizice a datelor: *Programele de aplicaţie şi activităţile utilizatorilor nu depind de modul de depunere a datelor sau de modul de acces la date*.
- **Regula 10 –** regula independenţei logice a datelor. *Programele de aplicaţie trebuie să fie transparente la modificările de orice tip efectuate asupra datelor*.
- **Regula 11** – regula **independenţei datelor din punct de vedere al integrităţii**. *Regulile de integritate trebuie să fie* ***definite într-un sublimbaj relaţional, nu în programul de aplicaţie***.
- **Regula 12** – regula **independenţei datelor din punct de vedere al distribuirii**. *Distribuirea datelor pe mai multe calculatoare dintr-o reţea de comunicaţii de date nu trebuie să afecteze programele de aplicaţie*.
- **Regula 13 –** regula versiunii procedurale a unui SGBD. *Orice componentă procedurală a unui SGBD trebuie să respecte aceleaşi restricţii de integritate ca şi componenta relaţională.*

Regulile lui Codd

- Deoarece regulile lui Codd sunt prea severe pentru a fi respectate de un SGBD operaţional, s-au formulat criterii minimale de definire a unui sistem de gestiune relaţional.
- Un SGBD este **minimal relaţional** dacă:
  - toate datele din cadrul bazei sunt reprezentate prin **valori în tabele**;
  - nu există pointeri observabili de către utilizator;
  - sistemul suportă operatorii relaţionali de **proiecţie, selecţie şi compunere naturală**, fără limitări impuse din considerente interne.

- Un SGBD este **complet relaţional** dacă este minimal relaţional şi satisface în plus condiţiile:
  - sistemul suportă **restricţiile de integritate de bază** (unicitatea cheii primare, constrângerile referenţiale, integritatea entităţii).
  - sistemul suportă **toate operaţiile de bază ale algebrei relaţionale**.

NORMALIZAREA RELAȚIILOR

- În procesul modelării unei baze de date relaţionale, o etapă importantă o reprezintă **normalizarea relaţiilor conceptuale** (Codd), adică obţinerea de relaţii „moleculare“ fără a pierde nimic din informaţie pentru a elimina:
  - **redundanţa**;
  - **anomaliile reactualizării informaţiilor**.

NORMALIZAREA RELAȚIILOR

- Tehnica normalizării permite:
  - obţinerea unei **scheme conceptuale rafinate** printr-un proces de **ameliorare progresivă** a unei scheme conceptuale iniţiale a bazei de date relaţionale.
- După fiecare etapă de ameliorare, relaţiile bazei de date ating un anumit **grad de perfecţiune**, deci se află într-o anumită **formă normală**.
  - Trecerea unei relaţii dintr-o formă normală în alta, presupune **eliminarea unui anumit tip de dependenţe** nedorite, care sunt transformate în dependenţe admisibile, adică dependenţe care nu provoacă anomalii.

NORMALIZAREA RELAȚIILOR

- Procesul de ameliorare a schemei conceptuale **trebuie**:
  - să garanteze **conservarea datelor**, adică în schema conceptuală finală trebuie să figureze toate datele din cadrul schemei iniţiale;
  - să garanteze **conservarea dependenţelor** dintre date, adică în schema finală fiecare dependenţă trebuie să aibă determinantul şi determinatul în schema aceleiaşi relaţii;
  - să reprezinte o **descompunere minimală** a relaţiilor iniţiale, adică niciuna dintre relaţiile care compun schema finală nu trebuie să fie conţinută într-o altă relaţie din această schemă.

NORMALIZAREA RELAȚIILOR

Există două metode pentru a modela **baze de date relaţionale fără anomalii sau pierderi de informaţie**.

- **Schema descompunerii** pleacă de la o **schemă relaţională universală** ce conţine toate atributele BD.
  - Schema se descompune prin proiecţii succesive în subrelaţii.
  - Descompunerea se opreşte când continuarea ei ar duce la pierderi de informaţie.
  - Algoritmii de descompunere se bazează, în general, pe descrierea formală a dependenţei dintre atribute.
- **Schema sintezei** pleacă de la o mulţime de atribute independente.
  - Utilizând proprietăţi de semantică şi legături între atribute se pot compune noi relaţii, astfel încât, acestea să nu sufere de anumite anomalii.
  - Algoritmii se bazează, în general, pe teoria grafurilor pentru a reprezenta legăturile între atribute.

Dependenţe funcţionale

- O **relaţie universală** este o relaţie ce grupează toate atributele care modelează sistemul real cercetat.
- Fie *E*, mulţimea dependenţelor considerate de proiectantul bazei pentru o schemă relaţională sau pentru o relaţie universală.
  - Plecând de la o mulţime de proprietăţi formale ale dependenţelor, proprietăţi considerate drept reguli de deducţie (axiome), poate fi obţinută mulţimea maximală de dependenţe asociate lui *E*. Această mulţime defineşte **închiderea** lui *E*.

Dependenţe funcţionale

- Fie E mulţimea dependenţelor unei relaţii şi $p_1, p_2, ..., p_r$, $r \geq 1$, proprietăţi formale ale acestor dependenţe.
  - Dacă există o mulţime E', astfel încât orice dependenţă a mulţimii E este derivabilă din E' prin aplicarea proprietăţilor $p_1, p_2, ..., p_r$, atunci mulţimea E' defineşte **acoperirea** lui E pentru proprietăţile $p_1, p_2, ..., p_r$.
- E' este o **acoperire minimală** pentru E, dacă nu există nici o submulţime proprie, nevidă a lui E' care să fie o acoperire pentru E. **Evident, E şi E' au închideri identice, deci dispun de acelaşi potenţial informaţional!**

Dependenţe funcţionale

- Fie $R(A_1, A_2, ..., A_n)$ o schemă relaţională şi fie X, Y submulţimi de atribute ale lui R.
  - X **determină funcţional** Y sau Y depinde funcţional (FD) de X, dacă pentru orice relaţie r (valoare curentă a lui R) nu există două tupluri care să aibă aceleaşi valori pentru atributele lui X şi să aibă valori diferite pentru cel puţin un atribut din Y. Cu alte cuvinte, o valoare a lui X, determină unic o valoare a lui Y.
- Notaţie: $X \rightarrow Y$. X este numit **determinant**, iar Y este numit **determinat** (sau dependent).
  - Dependenţa funcţională $X \rightarrow Y$ este **trivială** dacă $Y \subseteq X$.

Dependenţe funcţionale

- Comparând toate submulţimile de atribute ale unei relaţii şi determinând legăturile dintre ele, se pot obţine toate dependenţele funcţionale pe care o relaţie le satisface.
  - Această abordare **nu** este eficientă, consumând mult timp.
- Există posibilitatea ca, ştiind **anumite dependenţe** funcţionale şi utilizând **reguli de deducţie**, să fie obţinute **toate** dependenţele funcţionale.

Dependenţe funcţionale

Fie X, Y, Z, W mulţimi de atribute ale unei scheme relaţionale R şi fie următoarele axiome:

- **Ax1 –** reflexivitate. $X \to X$. Mai general, dacă $Y \subseteq X$, atunci $X \to Y$.
- **Ax2 –** creşterea determinantului. Pot fi considerate următoarele formulări echivalente pentru această axiomă.
  - Dacă $X \to Y$ şi $X \subseteq Z$, atunci $Z \to Y$.
  - Dacă $X \to Y$ şi $W \subseteq Z$, atunci $X \cup Z \to Y \cup W$.
  - Dacă $X \to Y$ atunci $X \cup Z \to Y \cup Z$.
  - Dacă $X \to Y$ atunci $X \cup Z \to Y$.
- **Ax3** – tranzitivitate. Dacă $X \to Y$ şi $Y \to Z$, atunci $X \to Z$.

Dependenţe funcţionale

- O mulţime de axiome este **completă** dacă şi numai dacă plecând de la o mulţime de dependenţe *E* se pot obţine toate dependenţele închiderii lui *E*, utilizând axiomele mulţimii.
- O mulţime de axiome este **închisă** dacă şi numai dacă plecând de la o mulţime de dependenţe *E*, nu poate fi dedusă cu ajutorul axiomelor o dependenţă care nu aparţine închiderii lui *E.* (nu obţin altele!)
- **Ullman** a demonstrat că axiomele Ax1 – Ax3, numite axiomele lui Amstrong, reprezintă o mulţime închisă şi completă de axiome. Consecinţa acestui rezultat este că **închiderea lui *E* reprezintă mulţimea dependenţelor deduse din *E*, prin aplicarea axiomelor lui Amstrong!!!**

Dependenţe funcţionale

- Nu toate dependenţele funcţionale sunt folositoare pentru modelarea relaţională.
- O dependenţă funcţională $X \rightarrow Y$ se numeşte **dependenţă funcţională totală** (FT), dacă şi numai dacă nu există nicio submulţime proprie $X' \subset X$, astfel încât $X' \rightarrow Y$.
  - Dacă există o submulţime proprie $X' \subset X$, astfel încât $X' \rightarrow Y$, atunci dependenţa funcţională $X \rightarrow Y$ este **parţială**. În axioma Ax2, dependenţa $Z \rightarrow Y$ este o dependenţă funcţională parţială.

Dependenţe funcţionale

- În cazul dependenţei funcţionale totale, axiomele lui Amstrong se reduc la o axiomă unică şi anume **pseudo-tranzitivitatea**:
  - **dacă $X \rightarrow Y$ şi $W \cup Y \rightarrow Z$, atunci $W \cup X \rightarrow Z$.**
- Această axiomă este o regulă de deducţie completă pentru total dependenţe:
  - pseudo-tranzitivitatea implică tranzitivitatea ($W = \emptyset$);
  - reflexivitatea nu poate fi utilizată pentru a obţine dependenţe totale;
  - reflexivitatea şi pseudo-tranzitivitatea implică creşterea.

Dependenţe funcţionale

- Dacă *F* este o mulţime de dependenţe funcţionale totale, atunci **închiderea pseudo-tranzitivă** F^+ a acestei mulţimi este reuniunea mulţimilor dependenţelor funcţionale totale care pot fi obţinute din *F* folosind axioma de pseudo-tranzitivitate.
- Două mulţimi de dependenţe funcţionale totale sunt **echivalente** dacă au închideri pseudo-tranzitive identice.
  - Pentru a modela scheme relaţionale se consideră **mulţimi minimale de dependenţe funcţionale totale**, capabile să genereze toate închiderile pseudo-tranzitive. Aceste mulţimi definesc **acoperiri minimale**.

Dependenţe funcţionale

- O mulţime de dependenţe funcţionale totale F^* asociată unei mulţimi de atribute A defineşte o **acoperire minimală** dacă satisface următoarele proprietăţi:
  - nici o dependenţă funcţională din F^* nu este redundantă;
  - toate dependenţele funcţionale totale între submulţimi ale lui A sunt în închiderea pseudo-tranzitivă a lui F^*.
- Orice mulţime de dependenţe totale are cel puţin o acoperire minimală. Alegerea acoperirii minimale este punctul de start în modelarea schemelor relaţionale.

Dependenţe funcţionale

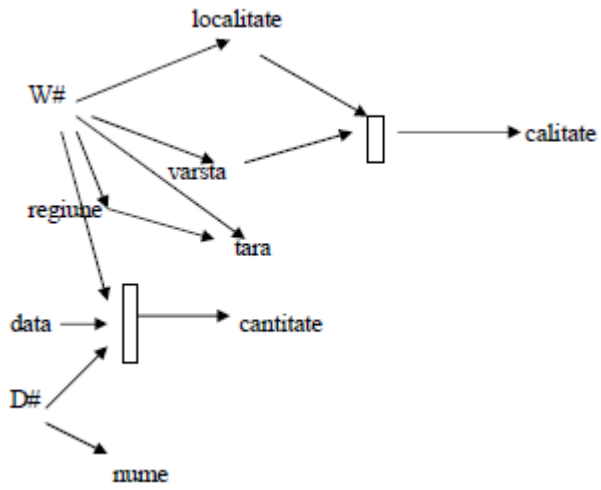
- Dependenţele funcţionale între atributele bazei pot fi reprezentate grafic.
  - Fie $A = \{A_1, A_2, ..., A_n\}$ o mulţime de atribute şi fie o mulţime de dependenţe funcţionale $\{X_i \rightarrow A_j\}$, unde X_i este o submulţime a lui A.
- **Graful dependenţelor funcţionale** este un graf direcţionat bipartit, definit astfel:

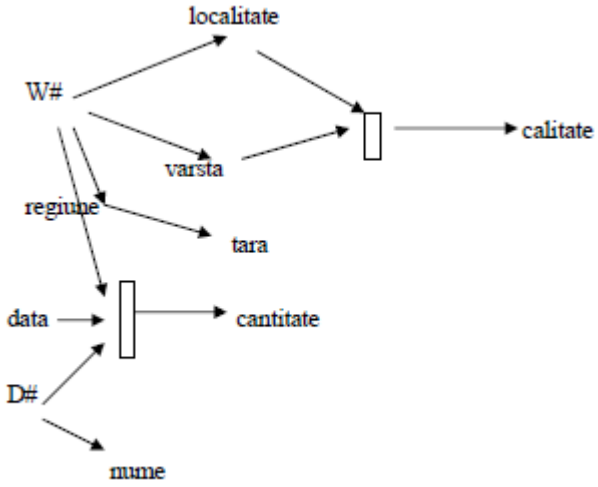
1. pentru fiecare atribut A_j există un singur **nod** având eticheta A_j;
2. pentru fiecare dependenţă funcţională de forma $A_i \rightarrow A_j$, există un **arc** de la A_i la A_j;
3. pentru fiecare dependenţă funcţională de forma $X_i \rightarrow A_j$, unde mulţimea X_i este definită de $X_i = \{Ai_1, ..., Ai_p\}$ cu $p > 1$, există un **nod auxiliar** etichetat prin X_i şi **o mulţime de arce** plecând de la $Ai_1, ..., Ai_p$ pentru a obţine pe X_i şi **printr-un arc adiţional** de la X_i la A_j. Nodurile X_i se reprezintă prin dreptunghiuri.

Dependenţe funcţionale

Exemple:

1. Graful dependenţelor funcţionale pentru schema relaţională CONSUMATOR_DE_VIN(W#, localitate, varsta, calitate, regiune, tara, D#, nume, data, cantitate) şi acoperirea minimală.





Dependenţe funcţionale

2. Graful dependenţelor funcţionale pentru schema relaţională OBIECTIV_INVESTITIE.
Dependențele sunt deduse din regulile impuse de beneficiar!