CONDIȚII SUFICIENTE PENTRU SCHEME ÎN FORMELE NORMALE TREI ŞI BOYCE-CODD

V. Cotelea, dr. conf. univ.
Academia de Studii Economice din Moldova

Fie $Sch = (R, F)$ o schemă a bazei de date, unde F este o mulțime de dependențe funcționale definite pe o mulțime de atribute R. Mulțimea de dependențe funcționale F poate avea diversă structură. În primul rând, poate fi o mulțime neredundantă, în al doilea rând, poate fi mulțime redusă, împărțită în clase de echivalență sau poate fi minimală. [1]

Două mulțimi de dependențe funcționale F şi G (se notează $F \equiv G$) sunt echivalente, dacă şi numai dacă $F^+ = G^+$, adică, dacă închiderile acestor mulțimi sunt aceleaşi.

Mulțimea de dependențe funcționale F se numeşte neredundantă [2], dacă $\nexists G$, astfel că $G \subset F$ şi $G \equiv F$.

Fie că prin $|F|$ şi prin $\|F\|$ sunt notate cardinalitatea mulțimii F şi numărul de atribute antrenate de F (inclusiv cele repetate), respectiv.

Mulțimea de dependențe funcționale F se numeşte minimală [3], dacă $\nexists G$, astfel că $G \equiv F$ şi $|G| < |F|$.

Fie F o mulțime de dependențe funcționale asupra schemei R şi $X \to Y \in F$. Atributul A este redundant în dependența $X \to Y$ în raport cu F, dacă
$A \in X$, $F - \{X \to Y\} \cup \{(X - \{A\}) \to Y\} \equiv F$ sau
$A \in Y$, $F - \{X \to Y\} \cup \{X \to (Y - \{A\})\} \equiv F$.

Mulțimea F se numeşte redusă în stânga (dreapta), dacă orice dependență din F nu are atribute redundante în partea stângă (dreaptă). Mulțimea de dependențe redusă în stânga şi în dreapta se numeşte redusă.

Fie $X \to Y \in F$. Se defineşte în calitate de clasă de echivalență de dependențe funcționale mulțimea de dependențe, din care face parte şi $X \to Y$, notată $E_F(X)$:

$$E_F(X) = \{V \to W \mid V \to W \in F \wedge X \leftrightarrow V\}$$

Astfel, $E_F(X)$ este mulțimea de dependențe din F cu părțile stângi echivalente cu X în raport cu F.

În continuare, se presupune că mulțimea F de dependențe funcționale este redusă şi împărțită în clase de echivalență: $F = F_1 \cup ... \cup F_n$

Se va nota cu R_i mulțimea de atribute antrenate de dependențele clasei F_i, $i = \overline{1,n}$, şi se va nota cu $F - F_i(C) = (F - F_i) \cup \{X \to (Y - C) \mid X \to Y \in F_i\}$.

Fie $X \to Y \in F^+$ şi fie $< X_0, X_1, ..., X_n >$ derivarea maximală [4] a mulțimii X în raport cu F. Fie X_i este primul element din secvență ce conține mulțimea Y. Subsecvența $< X_0, X_1, ..., X_i >$ se numeşte *derivarea* (nu neapărat maximală) dependenței funcționale $X \to Y$ în raport cu F.

Definiția 1. [4] $X \to Y \in F^+$, dacă şi numai dacă există derivarea dependenței $X \to Y$ în raport cu F.

Aceste construcții şi afirmații pot fi folosite în determinarea gradului de normalizare a schemei bazei de date. Este cunoscut faptul că problema de determinare a gradului de normalizare este una de natură exponențială. În primul rând, definițiile schemelor normale (forma normală doi, trei sau Boyce-Codd) conțin noțiunea de cheie. Dar e cunoscut şi faptul că o relație poate avea un număr exponențial de chei în raport cu numărul de atribute ale schemei acesteia. În al doilea rând, definițiile formelor normale apelează la noțiunile de atribute primare şi neprimare, noțiuni iarăşi legate de noțiunea de cheie.

Mulțimea de atribute $K \subseteq R_i$ se numeşte cheie a schemei $Sch_i = (R_i, F)$, dacă satisface următoarele condiții:

(1) $K \to R_i \in F^+$

(2) $\forall K' \subset K$, $K' \to R_i \notin F^+$

Atributele care fac parte dintr-o cheie a schemei date se numesc primare, în caz contrar – neprimare.

Schema $Sch_i = (R_i, F)$ se găseşte în *forma normală trei* [5] în raport cu mulțimea de dependențe funcționale F, dacă ea se găseşte în forma normală unu şi orice atribut neprimar nu depinde tranzitiv de vreo cheie a schemei Sch_i. Schema bazei de date se găseşte în forma normală trei, dacă orice schemă relațională constituentă se găseşte în forma normală trei.

Fie schema $Sch_i = (R_i, F)$, unde $V, W \subseteq R_i$ şi $A \in R_i$. Se spune că atributul A *depinde tranzitiv* de V prin W, dacă sunt satisfăcute condițiile:

(1) $V \to W \in F^+$;

(2) $W \to V \notin F^+$ (adică V nu depinde funcțional de W);

(3) $W \to A \in F^+$;

(4) $A \notin VW$.

Schema $Sch_i = (R_i, F)$ se găseşte în *forma normală Boyce-Codd* [6] în raport cu mulțimea de dependențe funcționale F, dacă ea se găseşte în forma normală unu şi pentru orice dependență netrivială $V \to A \in F^+$ are loc $V \to R_i \in F^+$, adică partea stângă a fiecărei dependențe funcționale determină funcțional toate atributele schemei.

Definiția 2. [7] Fie clasa F_i, unde $|F_i| > 1$ şi $X \to YC \in F_i$. Atributul C este neesențial în schema $Sch_i = (R_i, F)$, dacă, pentru oricare două părți stângi V şi Z, unde $V, Z \in PS(F_i)$, are loc $V \to Z \in (F - F_i(C))^+$.

Definiția 3. [7] Atributul A, unde $A \in R_i$, este *recuperabil* în $Sch_i = (R_i, F)$, dacă $(R_i - A) \to A \in (F - F_i)^+$.

Fie schema relațională $Sch_i = (R_i, F)$ şi fie S şi T mulțimile de atribute neesențiale şi recuperabile, respectiv. Atunci, poate fi formulată următoarea aserțiune:

Lema 1. $S \cap T = \varnothing$.

Demonstrație. Se presupune contrariul: $S \cap T \neq \varnothing$, adică, există în mulțimea de atribute R_i un atribut C, care este concomitent neesențial şi recuperabil în R_i. Fie atributul C este un element al părții drepte a dependenței $X \to YC$ din clasa de echivalență F_i. Ținând cont de definiția atributului neesențial, expresia $X \to Z \in (F - F_i(C))^+$ are loc pentru orice parte stângă Z din mulțimea părților stângi $PS(F_i)$. Prin urmare, $X \to (R_i - C) \in (F - F_i(C))^+$. Din definiția atributului recuperabil urmează că $(R_i - C) \to C \in (F - F_i)^+$ şi din faptul că $(F - F_i) \subseteq (F - F_i(C))$, atunci, are loc $(R_i - C) \to C \in (F - F_i(C))^+$. Din $X \to (R_i - C) \in (F - F_i(C))^+$ şi $(R_i - C) \to C \in (F - F_i(C))^+$, urmează că $X \to C \in (F - F_i(C))^+$. Dar acest lucru contravine presupunerii că dependența $X \to YC$ este redusă.

Lema 2. Dacă atributul A din R_i depinde tranzitiv de vreo cheie a schemei $Sch_i = (R_i, F)$, atunci, A este recuperabil în R_i.

Demonstrație. Deoarece A depinde tranzitiv de vreo cheie a schemei $Sch_i = (R_i, F)$, fie X, atunci, există o mulțime $V \subseteq R_i$, astfel că $X \to V \in F^+$, $V \to A \in F^+$, $V \to X \notin F^+$ şi $A \notin XV$. Atunci, din definiția 1, există derivarea $H = \langle V_0, ..., V_m \rangle$ pentru dependența $V \to A$ în raport cu F. Deoarece $V \to X \notin F^+$, atunci, $X \subseteq V_m$ şi partea stângă a fiecărei dependențe utilizată în construirea lui H nu este echivalentă cu X. Astfel, $V \to A \in (F - F_i)^+$. Dar $V \subseteq (R_i - A)$ şi, prin urmare, $(R_i - A) \to A \in (F - F_i)^+$

Teorema 1. Fie schema $Sch_i = (R_i, F)$. Atunci,

a) dacă orice atribut neprimar este neesențial, Sch_i se găseşte în forma normală trei.

b) dacă orice atribut recuperabil este primar, Sch_i se găseşte în forma normală trei.

Demonstrație. Veracitatea afirmațiilor a) şi b) reies din definiția schemei în forma normală trei şi lemele 1 şi 2.

Afirmațiile enunțate în teorema 1 includ noțiunile de atribute primare şi nepimare. Dar, precum e menționat în [8], problema determinării faptului dacă atributele sunt primare sau nu este una NP-completă.

Din punct de vedere aplicativ, mai acceptabilă este afirmația enunțată de următoarea teoremă:

Teorema 2. Dacă fiecare atribut A din R_i nu este recuperabil în schema $Sch_i = (R_i, F)$, atunci, $Sch_i = (R_i, F)$ se găseşte în forma normală Boyce-Codd.

***Demonstrație*.** Fie $Sch_i = (R_i, F)$ nu se găseşte în forma normală Boyce-Codd. Adică, există cel puțin o dependență $V \to A \in F^+$, unde $VA \subseteq R_i$, $A \notin V$ şi V nu este super-cheie pentru $Sch_i = (R_i, F)$. Dar, în acest caz, A depinde tranzitiv de vreo cheie a schemei Sch_i şi în virtutea lemei 2, A este recuperabil. Am obținut o contradicție.

Consecința 1. Dacă fiecare atribut A din R_i care nu aparține niciunei părți stângi a dependențelor din clasa de echivalență F_i, nu este recuperabil în $Sch_i = (R_i, F)$, atunci $Sch_i = (R_i, F)$ se găsește în forma normală trei.

Aici, trebuie menționat că condițiile teoremei 2 și consecinței 1 garantează existența schemelor în formele normale Boyce-Codd și forma normală trei, respectiv.

Însă, nu orice schemă care se găsește în forma normală Boyce-Codd (forma normală trei) satisfac condițiile teoremei 2 (consecinței 1). De exemplu, fie $F = F_1 \cup F_2 \cup F_3 \cup F_4$, unde $F_1 = \{KL \to B\}$, $F_2 = \{A \to K\}$, $F_3 = \{CE \to L\}$, $F_4 = \{ADE \to B,\ \ AB \to CDE\}$. Cu toate că atributul B este recuperabil în $Sch_4 = (\{A, B, C, D, E\},\ \ F)$, totuși Sch_4 se găsește în forma normală Boyce-Codd.

Cu toate acestea, algoritmii „rapizi” bazați pe teorema 2 și consecința 1 pot fi utili pentru analiza schemelor bazelor de date.

În afară de aceasta, consecința 1 permite aplicarea pentru sinteza schemei bazei de date un algoritm destul de simplu.

Într-adevăr, fie este dată o mulțime minimală de dependențe funcționale, divizată în clase de echivalență $F = F_1 \cup ... \cup F_n$. O astfel de mulțime poate fi obținută în $O(\| F \|^2)$ operații [1].

Pasul 1. Se construiește mulțimea Z_i tuturor atributelor, care se găsesc în părțile drepte și nu se găsesc în părțile stângi ale dependențelor din clasa de echivalență curentă F_i.

Pasul 2. Din mulțimea Z_i se selectează următorul atribut A.

Pasul 3. Se verifică dacă are loc $(R_i - A) \to A \in (F - F_i)^+$. Dacă da, atunci se setează $R_i := R_i - \{A\}$ și $Z_i := Z_i - \{A\}$. Pașii 2-3 se execută până nu sunt examinate toate atributele din Z_i.

Pasul 4. Se substituie părțile drepte ale dependențelor din clasa de echivalență F_i cu R_i. Pașii 1-4 se execută pentru toate clasele de echivalență ale dependențelor funcționale.

Pasul 5. Mulțimea de dependențe funcționale se reduce.

Fiecare clasă de echivalență va reprezenta o schemă relațională, iar părțile stângi ale dependențelor din clasă vor reprezenta cheile posibile ale schemei.

Deoarece toate atributele recuperabile din părțile drepte ale dependențelor sunt eliminate, schema bazei de date obținută în urma aplicării pașilor descriși va fi constituită din scheme relaționale în forma normală trei.

Nu este greu de arătat că complexitatea acestui algoritm este aceeași ca și complexitatea algoritmului propus de Bernstein [9] - $O(\| F \|^2)$.

Bibliografie

1. **Maier, D.** *The theory of relational database. Computer Science Press, 637 p., 1983.*
2. **Paredaens, J.** *About Functional Dependencies in a Database Structure and their Coverings. Phillips MBLE Lab. Report 342, 1977.*
3. **Maier,. D.** *Minimum cover in the relational database model. Jour. Of ACM, V.27, N. 4, p 664…674, 1980.*
4. **Cotelea, Vitalie.** *An inference model for functional dependencies in database schemas. Meridian Ingineresc, N.3, p. 89...92, 2009.*
5. **Codd, E.F.** *Further Normalization of the data base relational model. Data Base Systems, R. Rustin (ed), Prentice Hall, p. 33…64, 1972.*
6. **Codd, E.F.** *Recent Investigation in Relation Data Base Systems. IFIP Congress, p. 1017…1021, 1974.*
7. **Cotelea, Vitalie.** *Atribute neesențiale și recuperabile în schemele bazelor de date. Meridian Ingineresc, N.2, p. 18…19, 2009.*
8. **Beeri, C., Bernstein, Philip A.** *Computational problems related to the design of normal form relational database. ACM Trans. Database Syst., V.301, N. 4, p.752…766, 1983*
9. **Bernstein, Philip A.** *Synthesizing Third Normal Form Relations from Funcțional Dependencies. ACM Trans. Database Syst., V.1, N. 4, p.277…298, 1976 .*

Recomandat spre publicare: 18.06.2009.