

1. Diagrama entitate/relatie si diagrama conceptuala (curs 2 - 5)
2. Arbore algebric si expresie algebrica (curs 6 - 10)
3. Forme normale (curs 6 - 10)
4. Dependente functionale (curs 6 - 10)

1. Diagrama entitate/relatie si diagrama conceptuala

Exercitiul 1:

O companie IT dezvolta mai multe proiecte, fiecare dintre acestea apartinand cate unui client. Compania are angajati, care pot fi de tip *administrator*, *dezvoltator*, sau *tester*. Fiecare proiect este condus de cate un singur angajat, care trebuie sa aiba rol de administrator. Un proiect poate avea mai multe versiuni. In procesul de testare a unei versiuni se observa anumite bug-uri. Un bug este raportat de catre un angajat de tip tester, un administrator stabileste un termen limita pentru rezolvarea bug-ului si il asigneaza unui dezvoltator. Un bug are o stare curenta(deschis, asignat, rezolvat, livrat, inchis). De-a lungul timpului bug-ul trece prin diferite stari. Un bug poate avea comentarii asociate, adaugate de catre dezvoltatori.

- a) Reprezentati diagrama conceptuala
- b) Reprezentati arborele si expresia algebrica pentru urmatoarea cerere: sa se afiseze informatii despre clientii care beneficiaza de proiecte pentru care au fost semnalate bug-uri dupa date de 01/06/2017. Rezolvarea este optima? Justificati.

2. Arbore algebric si expresie algebrica

Operatorii algebrei relationale sunt:

- operatori traditionali pe multimi (UNION, INTERSECT, PRODUCT, DIFFERENCE);
- operatori relationali speciali (PROJECT, SELECT, JOIN, DIVISION);

O expresie algebrica se poate reprezenta grafic cu ajutorul unui arbore algebric.

Etape :

Cerere -> arbore algebric (nu este unic) -> plan de executie -> optimizare

Un plan de executie reprezinta o secventa de pasi pentru evaluarea cererii precum si metoda care va fi folosita pentru evaluarea operatiei.

Exercitiu:

- 1. b)** Sa se afiseze informatii despre clientii care beneficiaza de proiecte pentru care au fost semnalate bug-uri dupa 01/06/2017.

```
select nume, prenume, telefon
from client c join proiect p on (c.id_client = p.id_client)
           join versiune v on (p.id_pr = v.id_pr)
           join bug b on (v.id_versiune = b.id_versiune)
where data_bug > to_date('01/06/2017', dd/mm/yyyy);
```

```
R1 = PROJECT (CLIENT, id_client, nume, prenume, telefon)
R2 = JOIN (R1, PROIECT)
R3 = JOIN (R2, VERSIUNE)
R4 = JOIN (R3, BUG)
R5 = SELECT (R4, data_bug > '01/06/2017')
REZ = PROJECT (R5, id_client, nume, prenume, telefon)
```

Arbore algebric:

Reguli de optimizare (curs 6 – 10, pagina 16)

3. Forme normale (curs 6 – 10, incepand cu pagina 24)

Formele normale:

FN1 – cere ca fiecare inregistrare sa fie definita astfel incat sa fie identificata in mod unic prin intermediul unei chei primare.

EX:

PERSOANA	JOCURI PREFERATE
Daniel	J1, J2
Matei	J2, J3

Nu este in FN1 -> este nevoie de normalizare

Normalizarea are drept scop:

- suprimarea redundantei logice;
- evitarea anomaliilor la reactualizare;
- rezolvarea problemei reconexiunii;

ID	PERSOANA	JOC PREF
1	DANIEL	J1
2	DANIEL	J2
3	MATEI	J2
4	MATEI	J3

FN2 - cere ca toate elementele unei tabele sa fie dependente functional de totalitatea cheii primare. Daca unul sau mai multe elemente sunt dependente functional numai de o parte a cheii primare, atunci ele trebuie sa fie separate in tabele diferite (la examen scrieti intreaga definitie din curs 6 – 10, paginile 28-29).

FN3 – fiecare atribut care nu este cheie trebuie sa depinda direct de cheie, de intreaga cheie primara. In exemplul din curs 6 – 10, pag 30, se observa cum x1 si x3 depind direct de cheia k, iar x2 depinde de cheie prin intermediul lui x1, ceea ce inseamna ca nu este in FN3 deoarece x2 nu depinde direct de cheie.

Boyce-Codd – este o versiune putin mai restrictiva de FN3. In cazul unei forme normale 3, toate attributele depind de o cheie, de intreaga cheie si numai de o cheie.

Tabelul de mai jos este in FN3 deoarece attributele depind numai de cheia primara, dar exista o redundanta deoarece perechile (cod_objectiv#, nr_contract) (O1, NR1), (O2, NR2) apar de cate doua ori in tabel. In acest caz coloanele cu date redundante se separa intr-un tabel (Vezi tabel 2)

cod_contractant#	cod_obiectiv#	nr_contract	cota_parte
C1	O1	NR1	20%
C1	O2	NR2	20%
C2	O2	NR2	30%
C2	O1	NR3	30%
C3	O1	NR1	35%

TABEL 1 – INVESTESTE_IN

INVESTESTE_IN(cod_contractant#, cod_obiectiv#, nr_contract, cota_parte).

Intre attributele relatiei exista dependentele:

{cod_contractant#, cod_obiectiv#} -> {nr_contract, cota_parte},
{nr_contract} -> {cod_obiectiv}.

Se aplica regula Casey-Delobel si se aduce relatia în BCNF.

INVESTESTE_IN_1(cod_obiectiv#, nr_contract);

INVESTESTE_IN_2(cod_contractant#, cod_obiectiv#, cota_parte);

cod_obiectiv	nr_contract#
O1	NR1
O2	NR2
O1	NR3

INVESTESTE_IN_1

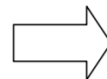
cod_contractant#	cod_obiectiv#	cota_parte
C1	O1	20%
C1	O2	20%
C2	O2	30%
C2	O1	30%
C3	O1	35%

INVESTESTE_IN_2

TABEL 2

FN4 – elimina redundantele datorate dependentei multiple. In urmatorul exemplu se observa cum un restaurant are mai multe tipuri de pizza si mai multe arii de distributie. Acest lucru duce la necesitatea inserarii mai multor intrari daca dorim, de exemplu, sa introducem un nou tip de pizza, fiind necesara cate o intrare pentru fiecare arie de distributie.

Restaurant	Tip de pizza	Aria de distributie
Pizza Hut	Clasic	Dr. Taberei
Pizza Hut	Clasic	Militari
Pizza Hut	Pufos	Dr. Taberei
Pizza Hut	Pufos	Militari
Springtime	Pufos	Domenii
Springtime	Clasic	Domenii
Jerry's	Clasic	Dr. Taberei
Jerry's	Clasic	Militari
Jerry's	Clasic	Crangasi
Jerry's	Pufos	Dr. Taberei
Jerry's	Pufos	Militari
Jerry's	Pufos	Crangasi



Restaurant	Tip de pizza
Pizza Hut	Clasic
Pizza Hut	Pufos
Springtime	Pufos
Springtime	Clasic
Jerry's	Clasic
Jerry's	Pufos

Restaurant	Aria de distributie
Pizza Hut	Dr. Taberei
Pizza Hut	Militari
Springtime	Domenii
Jerry's	Dr. Taberei
Jerry's	Militari
Jerry's	Crangasi

O descompunere fara pierdere de informatie, utilizata in procesul normalizarii, este data de **regula Casey-Delobel**:

Fie $R(A)$ o schema relationala si fie α, β, γ o partitie a lui A .

Presupunem ca α determina functional pe β . Atunci:

$$R(A) = \text{JOIN}(\Pi_{\alpha \cup \beta}(R), \Pi_{\alpha \cup \gamma}(R)).$$

$\alpha \cup \beta$ – multimea atributelor care intervin in dependentele functionale;

$\alpha \cup \gamma$ – reprezinta reuniunea determinantului cu restul atributelor lui A .

Explicatii FN2 si FN3

4. Dependente functionale. Graful dependentelor functionale (curs 6 – 10, paginile 20-24)

Exemplu:

Se considera relatia $R(\text{student}, \text{materie}, \text{profesor})$, in cadrul careia exista regulile:

- Pentru fiecare materie, fiecare student care o studiaza participa la cursurile unui singur profesor;
- Fiecare profesor predă numai o singura materie, dar o materie este predată de mai multi profesori;

Reprezentati graful dependetelor functionale. In ce forma normala se afla relatia? Normalizati.

Avem relatia: R(S#, nume, M#, an, semestru, P#, nume_prof, statut)

Schemele relationale:

STUDENT (id_stud#, nume)

MATERIE (id_m#, an, semestru)

PROFESOR (id_prof#, nume_prof, statut)

Trebuie extrase dependentele (relatiile):

M1 S1 P1 -> obligatoriu P1 sa fie numai cu M1 (adica acelasi profesor cu aceeasi materie), dar materia poate fi predata de mai multi profesori

M1 S2 P1

M2 S2 P2

M2 S6 P1 -> incorect deoarece P1 trebuie sa predea doar M1

M1 S1 P4 -> incorect deoarece un student trebuie sa mearga doar la cursurile unui singur profesor pentru o materie. Acelasi student, cu aceeasi materie, nu poate participa la un alt profesor

M2 S3 P2

M1 S3 P4 -> studentul la alta materie si alt profesor este bine

- Fiecare student participa la cursurile unui singur profesor
- Un profesor preda o singura materie, dar o materie este predada de mai multi profesori

