**Ministerul Educației și Cercetării al Republicii Moldova**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică și Microelectronică**

**Departamentul Ingineria Software și Automatică**

**Lucrare individuală**

*la disciplina* **”Baze de Date 2”**

*Tema:* **Sinteza schemei bazei de date în forma normală 3 (FN3)**

Efectuat de: studentul. gr. **TI-211** **Popa Cătălin**

Verificat de: asis. univ. **Grosu Olga**

**Chișinău 2024**

**Lucrare individuală**

La disciplina **’’Baze de date 2”**

Tema: **Sinteza schemei bazei de date în forma normală 3 (FN3)**

Este dată mulțimea de dependențe funcționale:

1. Să se construiască schema bazei de date în forma normală trei (3NF) utilizând algoritmul de sinteza.

**Pasul 1.** Se va adăuga la mulțimea de dependențe F dependența

**Pasul 2.** Se construiește acoperirea canonică a mulțimii de dependențe funcționale F

1. Se dezagregă dependențele funcționale din F care în partea dreaptă au mai mult decât un atribut:
2. Se construiește mulțimea neredundantă de dependențe funcționale:

D nu este submultime a lui (A)+, deaceea nu este redundanta in F.

F nu este submultime a lui (A)+, deaceea nu este redundanta in F.

G nu este submultime a lui (A)+, deaceea nu este redundanta in F.

C nu este submultime a lui (F)+, deaceea nu este redundanta in F.

A nu este submultime a lui (CG)+, deaceea nu este redundanta in F.

B este submultime a lui (CG)+, deaceea  **este redundanta in F.**

E nu este submultime a lui (CDG)+, deaceea nu este redundanta in F.

B nu este submultime a lui (A)+, deaceea nu este redundanta in F.

C este submultime a lui (AG)+, deaceea  **este redundanta in F**.

FG este submultime a lui (FG)+, deaceea  **este redundanta in F.**

}

1. Dependențele se reduc în stânga:

B nu este submultime a lui (C)+, deaceea nu este redundanta in F.

B nu este submultime a lui (G)+, deaceea nu este redundanta in F.

E nu este submultime a lui (CD)+, deaceea nu este redundanta in F.

E este submultime a lui (CG)+, deaceea  **este redundanta in F.**

E nu este submultime a lui (DG)+, deaceea nu este redundanta in F.

E nu este submultime a lui (C)+, deaceea nu este redundanta in F.

B nu este submultime a lui (D)+, deaceea nu este redundanta in F.

B nu este submultime a lui (G)+, deaceea nu este redundanta in F.

Q este submultime a lui (ABCDEF)+, deaceea  **este redundanta in F**.

Q este submultime a lui (ABCDE)+, deaceea  **este redundanta in F**.

Q este submultime a lui (ABCD)+, deaceea **este redundanta in F**.

Q este submultime a lui (ABC)+, deaceea  **este redundanta in F**.

Q nu este submultime a lui (AB)+, deaceea  **este redundanta in F.**

Q este submultime a lui (AC)+, deaceea **este redundanta in F**.

Q nu este submultime a lui (A)+, deaceea  **este redundanta in F.**

Q nu este submultime a lui (C)+, deaceea nu este redundanta in F.

**Pasul 3.** Împărțim mulțimea Flr în clase de echivalență Ef. Pentru a face acest lucru, construim închideri pentru toți determinanții din Flr:

S-au găsit următoarele clase de echivalență

Clasele de echivalență rezultate sunt:

O dependență care conține Q ca determinant este inclusă într-una dintre clasele de echivalență (E(CFG)), deci va fi exclusă. Clase de echivalență modificate:

**Pasul 4.** Construim o mulțime suplimentară J în următoarea ordine: inițial J=0. Pentru oricare două dependențe funcționale din F cu determinanții X și Y, unde , modificăm: . Drept urmare, obținem mulțimea.

Pentru orice A din Y, dacă este în F, atunci: ; Același lucru se face pentru orice B din X. Dacă este în F, atunci: . Clasele de echivalență rezultate sunt:

**Pasul 5.** Ștergem dependențele tranzitive din F. Există o submulțime F1 a lui F care îndeplinește condiția și nu există o submulțime propriu-zisă F1 care să nu îndeplinească această condiție. Pentru a face acest lucru, verificăm dacă F nu conține dependențe redundante.

**:**

D nu este submultime a lui (A)+, deaceea nu este redundanta in F.

**:**

C nu este submultime a lui (A)+, deaceea nu este redundanta in F.

**:**

C nu este submultime a lui (F)+, deaceea nu este redundanta in F.

**:**

B nu este submultime a lui (C)+, deaceea nu este redundanta in F.

**:**

E nu este submultime a lui (D)+, deaceea D nu este redundanta in F.

**:**

A nu este submultime a lui (FG)+, deaceea nu este redundanta in F.

Mulțimea F1:

Includem dependențele de la J în clasele de echivalență corespunzătoare ale mulțimii F1 și obținem mulțimea G.

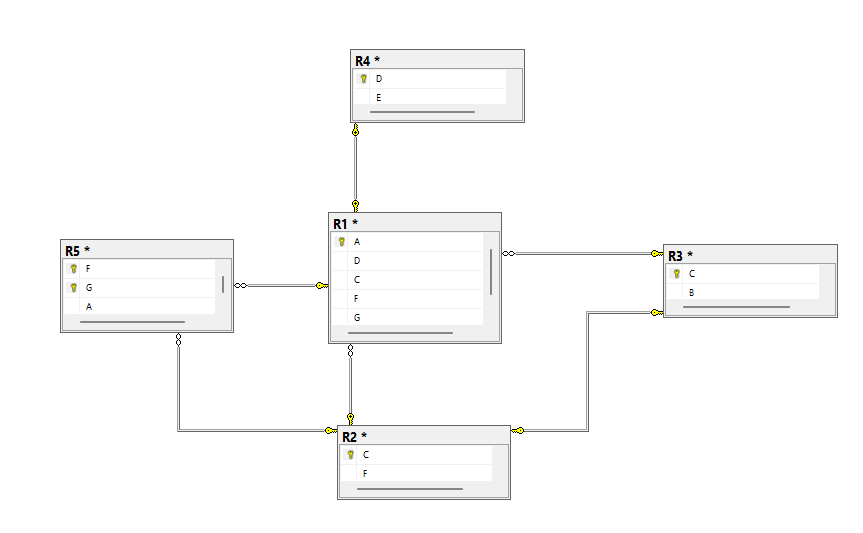
**Pasul 6.** Construim circuite R1, ... ,Rm. Fiecare schemă Ri conține atribute ale dependențelor funcționale din clasa a i-a de echivalență. Schemele de relații rezultate în a treia formă normală (3FN):

Schema de bază de date sintetizată în a treia formă normală (3NF):

*Bd = { (ADCFG, {A}), (CF, {F}), (CB, {C}), (DE, {D}) , (FGA, {FG}) }*

1. **Pentru schema sintezată a bazei de date construiți diagrama bazei de date.**

În baza rezultatelor obținute la sinteza bazei de date în forma normală 3 am obținut 4 relații a căror diagramă este reprezentată în figura 1.



**Figura 1 – Diagrama bazei de date**

1. Să se descrie în forma de text asocierile dintre relațiile diagramei obținute.

Conform diagramei de mai sus putem distinge doar relații unu la mulți. Ele sunt următoarele:

R1 și R2 – mulți la unu cu cheia externă C

R1 și R3 – mulți la unu cu cheia externă C

R1 și R5 – unu la multi cu cheia externă FG

R5 cu R2 – multi la unu cu cheia externa C

R1 cu R4 – unu la unu cu cheia externa D

R2 cu R3 – unu la unu cu cheia externa C

1. Pentru fiecare relație din BD să se indice care este cheia primară (să fie subliniate atributele primare în schemă), cheile secundare, cheile externe.

*Bd = { (ADCFG, {A}), (CF, {F}), (CB, {C}), (DE, {D}) , (FGA, {FG}) }*

Cheie primara – A

Cheie secundara –

Cheie externa – F,G,C,D

Cheie primara – F

Cheie secundara – --

Cheie externa – C

Cheie primara – C

Cheie secundara – --

Cheie externa – B

Cheie primara – G

Cheie secundara – --

Cheie externa – E

Cheie primara – FG

Cheie secundara – --

Cheie externa – A