Capitolul 4

PROIECTAREA BAZELOR DE DATE

PROBLEMATICA

- O categorie de probleme care pot sa apara in dezvoltarea unei aplicatii continand o baza de date este cea a proiectarii incorecte a schemelor de relatie.
- In acest caz pot sa apara o serie de ***anomalii*** care pot complica procesul de programare.
- Testarea corectitudinii unei scheme de relatie poate fi facuta cu ajutorul ***dependentelor functionale - DF*** (sau de alt tip) atasate acelei scheme.

PROBLEMATICA – CONT.

- DF modeleaza corelatii care exista intre datele din lumea reala stocate in baza de date si reprezinta criterii de corectitudine ale datelor incarcate in baza de date.
- In cazul in care o relatie nu are o schema corespunzatoare ea trebuie inlocuita cu doua sau mai multe relatii (operatia este numita si ***descompunerea unei scheme de relatie***), fiecare relatie rezultata avand o schema corecta – aflata in ***forma normala*** dorita.

ANOMALII (Cap 2)

IDP	NUMEP	QTY	IDF	NUMEF	ADRESAF
101	Imprimantă laser	30	20	XY SRL	Str. X, Bucureşti
105	Calculator PC	20	23	Z SRL	Bd. Z, Bucureşti
124	Copiator	10	20	XY SRL	Str. X, Bucureşti

ANOMALII (1)

- **Redundanta:** Redundanta reprezinta stocarea in mod nejustificata a unei aceleiasi informatii de mai multe ori in baza de date.
- Observam ca pentru fiecare produs este stocat numele si adresa furnizorului, desi ele sunt unic determinate de codul acestuia.

ANOMALII (2)

- **Anomalia de stergere:** La stergerea din relatie a ultimului produs al unui furnizor se pierd automat si datele despre acesta.

ANOMALII (3)

- **Anomalia de actualizare:** In cazul actualizarii unei informatii redundante, se poate intampla ca operatia sa modifice unele aparitii ale acesteia iar altele sa ramana cu vechea valoare.

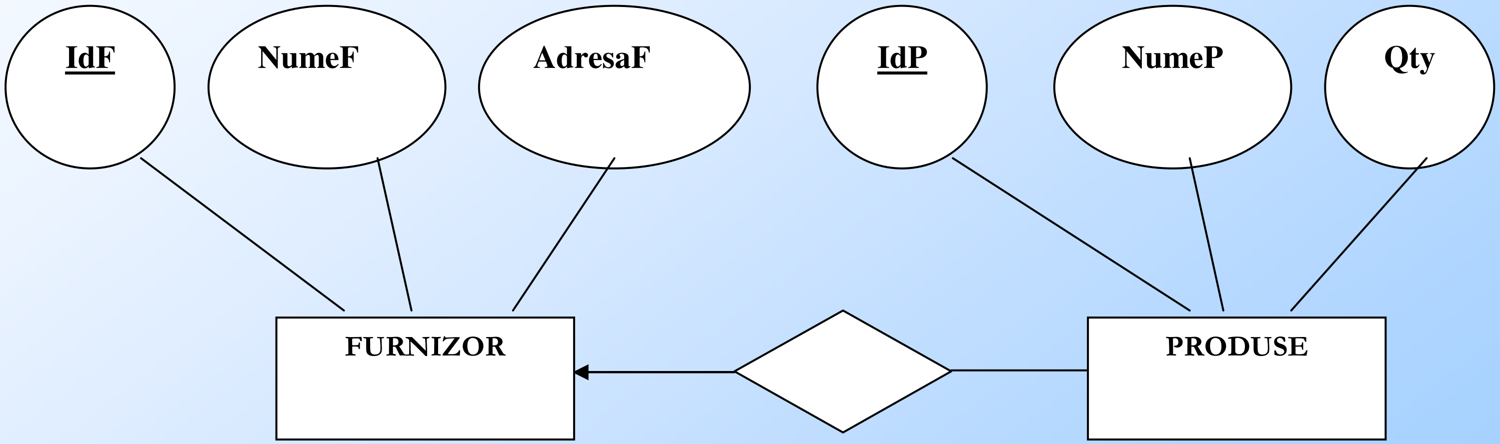
ANOMALII (4)

- **Anomalia de inserare:** Nu putem insera date despre un furnizor (numele si adresa sa) decat daca exista in stoc un produs furnizat de acesta.

SURSA ANOMALIILOR DIN EXEMPLU

- Aceste anomalii apar in relatia PRODUSE deoarece intr-o aceeasi tabela au fost stocate date despre doua clase diferite de obiecte.
- In cazul proiectarii cu ajutorul modelului entitate-asociere diagrama corecta este urmatoarea:

DIAGRAMA EA



REZULTAT TRANSFORMARE

Prin transformarea acestei diagrame se obtin urmatoarele scheme de relatie:

- Furnizor(<u>IdF</u>, NumeF, AdresaF)
- Produse(<u>IdP</u>, NumeP, Qty, IdF)

Tabelele FIRMA si PRODUS

IdF	NumeF	AdresaF
20	XY SRL	Str. X Bucureşti
23	Z SRL	Bd. Z, Bucuresti

IdP	NumeP	Qty	IdF
101	Imprimanta laser	30	20
105	Calculator PC	20	23
124	Copiator	10	20

OBIECTIVE DESCOMPUNERE

- Procesul de ‘spargere’ a unei tabele care are o structura incorecta in doua sau mai multe tabele se numeste ***descompunerea schemei de relatie***.
- Pentru detectarea relatiilor care trebuiesc descompuse exista o serie de reguli de corectitudine, numite si ***forme normale***.
- Definirea acestor forme normale se bazeaza pe notiunea de ***dependenta (functionala sau multivalorica)*** prezentata in continuare.

NOTAŢII (1)

- In paragrafele urmatoare vom folosi urmatoarea conventie de notare, intalnita in multe lucrari din literatura de specialitate a domeniului:
- R, S, T, …: scheme de relatii,
- r, s, …: instante ale relatiilor R respectiv S,

NOTAŢII (2)

- A, B, C, D, ... (litere mari de la inceputul alfabetului): atribute ale unei relatii,
- X, Y, Z, W, U, ... (litere mari de la sfarsitul alfabetului): multimi de atribute dintr-o schema de relatie,
- $X \subseteq R$: Multimea de atribute X este inclusa in multimea atributelor relatiei R.

NOTAŢII (3)

- $Y \subseteq X$: Multimea de atribute Y este inclusa in multimea de atribute X
- $A \in X$: Atributul A apartine multimii de atribute X
- t, t1, t2, … tupluri ale unei relatii,
- t[X]: valorile atributelor din X aflate in tuplul t,

NOTAŢII (4)

- F, G, ...: multimi de dependente functionale atasate unei scheme de relatie
- In paragrafele urmatoare termenul generic de **relatie** semnifica atat ***schema relatiei*** (descrierea structurii acesteia) cat si ***o instanta a acesteia*** (continutul de date de la un moment dat al relatiei).

DEPENDENŢE FUNCŢIONALE

Definitie: Fie:

- R o schema de relatie
- X, Y $\subseteq$R doua multimi de atribute ale acesteia.

Spunem ca ***X determina functional pe Y*** (sau Y este determinata functional de X) daca si numai daca oricare ar fi doua tupluri t1 si t2 din orice instanta a lui R atunci:

$$t1[X] = t2[X] \Rightarrow t1[Y] = t2[Y].$$

DEPENDENŢE FUNCŢIONALE (2)

- Altfel spus, daca doua tupluri au aceleasi valori pe atributele X atunci ele au aceleasi valori si pe atributele Y.
- Notatia pentru dependente functionale este o sageata de la stanga spre dreapta:

$$X \to Y$$

EXEMPLU

- **Exemplu:** In relatia Produse din paragraful anterior putem scrie urmatoarele dependente functionale:
- IdP $\rightarrow$ NumeP, Qty, IdF, NumeF, AdresaF,
- IdF $\rightarrow$ NumeF, AdresaF

Aceste dependente arata ca

- daca doua produse au acelasi IdP, este vorba de fapt de acelasi produs
- daca doua produse au acelasi IdF (Id furnizor) atunci si valorile pentru numele si adresa acestuia trebuie sa fie aceleasi.

OBSERVATIE IMPORTANTA

- Dependentele functionale nu se determina din inspectarea continutului de la un moment dat al relatiei ci din semnificatia atributelor acesteia.
- In exemplul prezentat, a doua DF arata ca daca la doua produse apare acelasi Id furnizor atunci numele si adresa furnizorului sunt de asemenea aceleasi (deoarece nu pot sa existe doi furnizori diferiti cu acelasi Id).

AXIOME SI REGULI

- Pornind de la o multime de dependente functionale atasate unei scheme de relatie se pot deduce alte dependente functionale valide.
- Exista o multitudine de reguli de inferenta. Pentru a se putea face o prezentare formala a acestora, trei dintre ele au fost alese ca axiome iar restul se pot deduce pornind de la ele.
- Cele trei axiome (numite in literatura si ***Axiomele lui Armstrong***) sunt urmatoarele:

A1 - REFLEXIVITATEA

- **A1. Reflexivitate:** Fie R o schema de relatie si $X \subseteq R$. Atunci:

  Daca $Y \subseteq X$ atunci $X \rightarrow Y$

- Toate dependentele functionale care rezulta din aceast axioma sunt numite si ***dependente triviale***. Ele nu spun nimic in plus fata de setul de dependente initial dar sunt dependente functionale valide.

A2 - AUGMENTARE

- **A2. Augmentare:** Fie R o schema de relatie si X, Y, Z $\subseteq$ R. Atunci:

  Daca X $\rightarrow$ Y atunci si XZ $\rightarrow$ YZ

- Aceasta axioma arata ca se poate reuni o aceeasi multime Z in stanga si in dreapta unei dependente functionale valide obtinand de asemenea o dependenta functionala valida.

A3 - TRANZITIVITATE

- **A3. Tranzitivitate:** Fie R o schema de relatie si X, Y, Z $\subseteq$ R.

  Daca X $\rightarrow$ Y si Y $\rightarrow$ Z atunci si X $\rightarrow$ Z

REGULI

- Pe baza acestor axiome se pot demonstra o serie de reguli de inferenta pentru dependente functionale dintre care cele mai importante sunt urmatoarele:

R1 - DESCOMPUNERE

- **R1. Descompunere:** Fie R o schema de relatie si X, Y, Z $\subseteq$ R.

  Daca X $\rightarrow$ Y si Z $\subseteq$ Y atunci si X $\rightarrow$ Z

- Regula descompunerii ne permite sa rescriem un set de dependente functionale astfel incat sa obtinem doar dependente care au in partea dreapta doar un singur atribut.

R1 - DESCOMPUNERE – cont.

- Sa presupunem ca avem o dependenta functionala de forma:

$$X \rightarrow A_1A_2A_3\ldots A_n$$

- Atunci ea poate fi inlocuita cu urmatoarele **n** dependente functionale:

$$X \rightarrow A_1$$
$$X \rightarrow A_2$$
$$X \rightarrow A_3$$
$$\ldots$$
$$X \rightarrow A_n$$

R2 - REUNIUNE

- **R2. Reuniune:** Fie R o schema de relatie si X, Y, Z $\subseteq$ R.

  Daca X $\rightarrow$ Y si X $\rightarrow$ Z atunci si X $\rightarrow$ YZ
- Rezulta si faptul ca din cele **n** reguli obtinute prin descompunere se poate obtine dependenta initiala, deci inlocuirea acesteia nu duce la pierderea vreunei corelatii existente.

R3 - PSEUDOTRANZITIVITATEA

- **R3. Pseudotranzitivitate:** Fie R o schema de relatie si X, Y, Z, W $\subseteq$ R.

Daca X $\rightarrow$ Y si YZ $\rightarrow$ W atunci si XZ $\rightarrow$ W

DEMONSTRATII

- **Exercitiu:** Demonstrati cele trei reguli folosind axiomele lui Armstrong.

Exemplu de demonstratie R3:

- Augmentam prima dependenta cu Z. Obtinem $XZ \rightarrow YZ$.
- Din aceasta dependenta si din $YZ \rightarrow W$ obtinem prin tranzitivitate $XZ \rightarrow W$, qed.

INCHIDEREA UNEI MULTIMI DE DF

- Pornind de la un set de dependente functionale F si utilizand axiomele si regulile obtinem o multitudine de alte dependente, triviale sau nu.
- Multimea tuturor dependentelor functionale care se pot deduce din F se numeste ***inchiderea multimii de dependente F***, notata cu F+.

DEFINITIE FORMALA

- Definitia formala a acestei inchideri este urmatoarea:

$$F+ = \{X \rightarrow Y \mid F \Rightarrow X \rightarrow Y \}$$

- Prin $\Rightarrow$ am notat faptul ca dependenta respectiva de poate deduce din F folosind axiomele si regulile.

OBSERVATIE

- Multimea F+ contine foarte multe dependente, inclusiv dependente triviale ca:
  - ABC $\rightarrow$ A,
  - ABC $\rightarrow$ B,
  - ABC $\rightarrow$ C,
  - ABC $\rightarrow$ AB,
  - ABC $\rightarrow$ AC,
  - ABC $\rightarrow$ BC sau
  - ABC $\rightarrow$ ABC

NU SE CALCULEAZA!

- Inchiderea unei multimi de dependente functionale nu se calculeaza, algoritmii care au nevoie de ea ocolind intr-un fel sau altul calculul acesteia.
- Introducerea acestei notiuni s-a facut pentru:
  - in cazul descompunerii unei scheme de relatie, aflarea dependentelor mostenite de la relatia initiala
  - pentru a putea defini formal alte notiuni

ACOPERIREA

- **Acoperirea unei multimi de DF:** Fie R o schema de relatie si F, G doua multimi de dependente pentru R. Se spune ca F ***acopera*** pe G daca si numai daca $G \subseteq F+$.

ECHIVALENTA

- **Echivalenta a doua multimi de dependente:**
- Fie R o schema de relatie si F, G doua multimi de dependente pentru R.
- Se spune ca F ***e echivalenta*** cu G daca si numai daca F acopera pe G si G acopera pe F (deci $G \subseteq F+$ si $F \subseteq G+$, deci $F+ = G+$)

FORMA CANONICA

- **Forma canonica a unei multimi de DF:**
- Din definitiile de mai sus rezulta ca o multime de dependente poate fi inlocuita cu alta echivalenta continand alte dependente.
- In cazul in care aceasta multime indeplineste conditiile urmatoare se spune ca este in ***forma canonica***:

FORMA CANONICA – cont.

- Orice dependenta are in partea dreapta un singur atribut. Acest lucru se poate obtine aplicand regula descompunerii prezentata anterior.
- Multimea de dependente este minimala, nici una dintre dependente neputand sa fie dedusa din celelalte (altfel spus nu exista dependente redundante).

EXEMPLUL 1

- Fie R = ABCDE o schema de relatie si F multimea de dependente functionale asociata, cu $F = \{ AB \rightarrow CDE, C \rightarrow DE \}$:
- Aplicam regula de descompunere. Obtinem:
- $F = \{ AB \rightarrow C, AB \rightarrow D, AB \rightarrow E, C \rightarrow D, C \rightarrow E \}$
- Noua F nu e minimala deoarece $AB \rightarrow D$ si $AB \rightarrow E$ se pot deduce prin tranzitivitate din $AB \rightarrow C$ impreuna cu $C \rightarrow D$, $C \rightarrow E$.
- Rezulta ca forma canonica a lui F este:

$$F = \{ AB \rightarrow C, C \rightarrow D, C \rightarrow E \}$$

EXEMPLUL 2

- Pentru relatia

Produse(IdP, NumeP, Qty, IdF, NumeF, AdresaF, IdF)

din paragraful 4.1. avand multimea de dependente functionale:

$F = \{ IdP \rightarrow NumeP, Qty, IdF, NumeF, AdresaF;$

$IdF \rightarrow NumeF, AdresaF\}$

EXEMPLUL 2 – cont.

- Forma canonica a lui F este:

$$F = \{ IdP \rightarrow NumeP,$$
$$IdP \rightarrow Qty,$$
$$IdP \rightarrow IdF,$$
$$IdF \rightarrow NumeF,$$
$$IdF \rightarrow AdresaF \}$$

EXEMPLUL 2 – cont.

Si in acest caz au fost eliminate doua dependente redundante:

- IdP $\rightarrow$ NumeF
- IdP $\rightarrow$ AdresaF

CHEIE

- **Definitie:** Fie R o schema de relatie, F multimea de dependente functionale asociata si $X \subseteq R$. Atunci X este cheie pentru R daca si numai daca:
  - $F \Rightarrow X \rightarrow R$ (deci $X \rightarrow R$ se poate deduce din F)

  si

  - X este minimala: oricare ar fi $Y \subset X$, $Y \neq X$ atunci $\neg(F \Rightarrow Y \rightarrow R)$ (deci orice submultime stricta a lui X nu mai indeplineste conditia anterioara).

CHEIE– cont.

- Deci: o cheie determina functional toate atributele relatiei si este minimala: nici o submultime stricta a sa nu determina functional pe R.
- Se observa faptul ca aceasta definitie este echivalenta cu cea din capitolul 3: cunoscandu-se valorile pe atributele X sunt unic determinate valorile pentru toate atributele relatiei, deci este unic determinat tuplul din relatie.

SUPERCHEIE

- In cazul in care doar prima conditie este indeplinita multimea X se numeste **supercheie**.
- **Observatie:** Faptul ca o supercheie nu este constransa de minimalitate nu inseamna insa ca ea nu poate fi minimala.
- Rezulta ca orice cheie este in acelasi timp si supercheie, reciproca nefiind insa adevarata.

EXEMPLU

- Fie R = ABCDE si F = { $AB \to C$, $C \to D$, $C \to E$ }. Atunci AB este cheie pentru R:
  - Din $AB \to C$, $C \to D$ si $C \to E$ obtinem prin tranzitivitate $AB \to D$ si $AB \to E$
  - Din $AB \to C$, $AB \to D$ si $AB \to E$ obtinem prin reuniune $AB \to CDE$
  - Din $AB \to CDE$ obtinem (augmentare cu AB) $AB \to ABCDE$, deci $AB \to R$
- Rezulta ca AB este supercheie pentru R. In paragraful urmator vom vedea cum se poate demonstra si faptul ca AB este minimala, deci este nu numai supercheie ci chiar cheie pentru R.

PROIECTIA UNEI MULTIMI DE DEPENDENTE FUNCTIONALE

- Asa cum s-a mentionat anterior inchiderea unei multimi de dependente functionale F^+ a fost introdusa si pentru a putea defini setul de dependente functionale mostenite de o schema de relatie obtinuta prin descompunerea unei scheme incorect proiectata.

PROIECTIA ... DF (2)

- Sa luam cazul relatiei anterioare continand produsele dintr-un depozit:

Produse = IdP, NumeP, Qty, IdF, NumeF, AdresaF

- avand asociata multimea de dependente:

F = { IdP→NumeP, IdP→Qty, IdP→ IdF, IdF→NumeF, IdF→AdresaF }

PROIECTIA ... DF (3)

- Prin descompunerea acestei relatii in doua obtinem relatiile:

Produse = IdP, NumeP, Qty, IdF

Furnizori= IdF, NumeF, AdresaF

- Atributele relatiei initiale se regasesc fie doar intr-una dintre schemele rezultate fie in amandoua. Se pune insa si problema: ce dependente mostenesc cele doua relatii de la relatia initiala?

PROIECTIA ... DF (3)

- Solutia este de a defini proiectia unei multimi de dependente pe o multime de atribute.
- **Definitie.** Fie o relatie R, o multime asociata de dependente functionale F si o submultime de atribute $S \subseteq R$. ***Proiectia multimii de dependente F pe S***, notata cu $\pi_S(F)$ este multimea dependentelor din F^+ care au si partea stanga si pe cea dreapta incluse in S.
- Formal putem scrie:

$$\pi_S(F) = \{X \rightarrow Y \in F^+ \mid X, Y \subseteq S \}$$

EXEMPLU

Pentru exemplul de mai sus proiectiile sunt urmatoarele:

- $F_{PRODUSE} = \pi_{PRODUSE}(F) =$

$\{ IdP \to NumeP, IdP \to Qty, IdP \to IdF \}$

- $F_{FURNIZORI} = \pi_{FURNIZORI}(F) =$

$\{ IdF \to NumeF, IdF \to AdresaF \}$

OBSERVATIE

- **Observatie:** Atunci cand descompunem o schema se poate intampla ca unele dintre dependentele schemei initiale sa se piarda.
- Exemplu: Fie R = ABCD si F = { $A \rightarrow B$, $B \rightarrow C$, $C \rightarrow D$, $D \rightarrow A$ }. In cazul in care descompunem R in R1 = AB si R2 = CD atunci:

  $F_{R1} = \pi_{R1}(F) = \{ A \rightarrow B, B \rightarrow A \}$

  $F_{R2} = \pi_{R2}(F) = \{ C \rightarrow D, D \rightarrow C \}$
- A doua dependenta din fiecare multime nu este in F dar este in F^+ (obtinuta prin tranzitivitate).
- Observam insa ca dependentele $B \rightarrow C$ si $D \rightarrow A$ nu mai pot fi obtinute nici din F_{R1} nici din F_{R2} nici din reuniunea lor.

INCHIDEREA UNEI MULTIMI DE ATRIBUTE

- Fie R o schema de relatie, F multimea de dependente asociata si $X \subseteq R$. Se poate defini ***inchiderea multimii de atribute X in raport cu F*** (notata X^+) astfel:
- $X^+ = \{ A \mid X \rightarrow A \in F^+ \}$
- X^+ contine deci toate atributele care apar in partea dreapta a unei dependente din F sau care se poate deduce din F folosind regulile si axiomele.

ALGORITM DE CALCUL X^+

- Intrare: R o schema de relatie, F multimea de dependente asociata si $X \subseteq R$
- Iesire: X^+
- Metoda: se procedeaza iterativ astfel:
  - Se porneste cu $X^{(0)} = X$
  - Pentru $i \geq 1$, $X^{(i)} = X^{(i-1)} \cup$

    $$\{ A \mid (\exists)\ Y \rightarrow A \in F \text{ cu } Y \subseteq X^{(i-1)} \}$$
  - Oprirea se face atunci cand $X^{(i)} = X^{(i-1)}$
  - Rezultat: $X^+ = X^{(i)}$

EXEMPLU

- Fie R = ABCDE si F = { $A \rightarrow B$, $A \rightarrow C$, $D \rightarrow E$ }. Pentru a calcula A^+ si AD^+ procedam astfel:

Calcul A^+ :

- $X^{(0)} = \{A\}$
- Din $A \rightarrow B$ si $A \rightarrow C$ rezulta ca $X^{(1)} = X^{(0)} \cup \{ B, C \} = \{ A \} \cup \{ B, C \} = ABC$
- Singurele dependente care au partea dreapta in $X^{(1)}$ sunt tot primele doua deci

  $X^{(2)} = X^{(1)} \cup \{ B, C \} = \{ A, B, C \} \cup \{ B, C \} = ABC$
- Oprire deoarece $X^{(2)} = X^{(1)}$
- Rezulta ca $(A)^+ = ABC$

EXEMPLU – cont.

Calcul AD⁺ :

- $X^{(0)} = \{A, D\}$
- Din $A \rightarrow B$, $A \rightarrow C$ si $D \rightarrow E$ rezulta ca $X^{(1)} = X^{(0)} \cup \{ B, C, E \} =$

  $\{ A, D \} \cup \{ B, C , E\} = ABCDE$
- Oprire deoarece $X^{(1)} = R$ deci oricate iteratii am face nu mai pot sa apara noi atribute.
- Rezulta ca $(AD)^+ = ABCDE$

INCHIDEREA … ATR. –cont.

- Scopul introducerii acestei notiuni este si cel de a putea ocoli in alti algoritmi si definitii calculul lui F^+ . Avem urmatorul rezultat teoretic:
- **Propozitie:** Fie R o schema de relatie, F multimea de dependente asociata si X, Y $\subseteq$ R. Atunci X $\rightarrow$ Y se poare deduce din F daca si numai daca Y $\in$ X^+
- Demonstratia acestei propozitii se gaseste in literatura de specialitate.

ALTA DEFINITIE A CHEII

- Pe baza propozitiei din paragraful anterior se poate da o alta definitie pentru cheia sau supercheia unei relatii, bazata nu pe F^+ ca in paragraful 4.2.5 ci pe inchiderea unei multimi de atribute.

CHEIE - REAMINTIRE

- **Definitie:** Fie R o schema de relatie si $X \subseteq R$. Atunci X este cheie pentru R daca si numai daca:
  - $F \Rightarrow X \rightarrow R$ (deci $X \rightarrow R$ se poate deduce din F)

  si

  - X este minimala: oricare ar fi $Y \subset X$, $Y \neq X$ atunci $\neg(F \Rightarrow Y \rightarrow R)$ (deci orice submultime stricta a lui X nu mai indeplineste conditia anterioara).

ALTA DEFINITIE A CHEII (2)

- **Definitie:** Fie R o schema de relatie, F multimea de dependente functionale asociata si $X \subseteq R$. Atunci X este cheie pentru R daca si numai daca:
  - $X^+ = R$

    si
  - X este minimala: oricare ar fi $Y \subset X$, $Y \neq X$ atunci $Y^+ \neq R$ (deci orice submultime stricta a lui X nu mai indeplineste conditia anterioara).
- Daca numai prima conditie este indeplinita atunci X este supercheie pentru R

ECHIVALENTA DEFINITII

Echivalenta acestei definitii cu cea anterioara este evidenta:

- $X^+ = R$ inseamna cf. propozitiei ca

  $X \rightarrow R$

- minimalitatea este de asemenea definita echivalent: $\neg(F \Rightarrow Y \rightarrow R)$ este echivalenta cu $\neg(Y^+ = R)$ adica $Y^+ \neq R$

GASIREA CHEILOR

- Folosind aceasta definitie se poate defini o euristica de gasire a cheilor unei relatii.
- Se cauta multimi minimale X care indeplinesc conditia $X^+ = R$
- Prezentam o euristica de gasire a cheilor:

NOTA IMPORTANTA

- Observatie: Atributele care nu apar in partea dreapta a nici unei dependente trebuie sa existe in orice cheie, ele neputand sa apara in procesul de calcul al inchiderii unei multimi de atribute.

EURISTICA

- Intrare: R o schema de relatie si F multimea de dependente functionale asociata (F in forma canonica).
- Iesire: Cheia unica sau cheile alternative ale lui R
- Metoda:

1. Se porneste de la multimea de atribute X $\subseteq$ R care nu apar in partea dreapta a nici unei dependente

EURISTICA (2)

2. Se calculeaza X^+. Daca $X^+ = R$ atunci X este cheia unica minimala a relatiei R si calculul se opreste aici. Pasii urmatori se efectueaza doar daca $X^+ \neq R$
3. Se adauga la X cate un atribut din $R - X^+$ obtinandu-se o multime de chei candidat.
4. Se calculeaza X^+ pentru fiecare dintre candidate. Daca se obtin toate atributele lui R atunci acel X este o cheie a lui R.

EURISTICA (3)

5. Se repeta pasii 3 si 4 pornind de la acele multimi candidat X care nu sunt gasite ca si chei la pasul anterior. Intre multimile candidat nu luam niciodata in considerare pe cele care contin o cheie gasita anterior.
6. Procesul se opreste cand nu se mai pot face augmentari.

EXEMPLUL 1

- R = ABCDE si
- F = { $A \rightarrow B$, $A \rightarrow C$, $D \rightarrow E$ }.
  - Multimea atributelor care nu apar in partea dreapta a nici unei dependente este X = AD.
  - Calculam $(AD)^+$. Obtinem (AD)+ = ABCDE = R.
  - Procesul se opreste. Rezulta ca AD este cheie unica pentru R

EXEMPLUL 2

- R = ABCDE si
- F = { A $\rightarrow$ B, B $\rightarrow$ A , A $\rightarrow$ C, D $\rightarrow$ E }.
  - Multimea atributelor care nu apar in partea dreapta a nici unei dependente este X = D.
  - Calculam $(D)^+$. Obtinem $(D)^+ = DE \neq R$. Rezulta ca D nu este cheie unica pentru R

EXEMPLUL 2 - cont

- $D^+ = DE$
  - Calculam multimea de candidate: augmentam D cu atribute din $R - D^+ = ABCDE - DE = ABC$. Obtinem AD, BD si CD
  - Calculam inchiderile lor. Obtinem $(AD)^+ = R$, $(BD)^+ = R$ si $(CD)^+ = CDE \neq R$. Rezulta ca AD si BD sunt chei ale lui R dar CD nu e cheie.

EXEMPLUL 2 – cont.

- Calculam o noua multime de candidate pornind de ca CD. Putem augmenta CD cu atribute din R – $(CD)^+$ = ABCDE – CDE = AB. Nici una dintre augmentari nu este insa posibila pentru ca atat ACD cat si BCD contin o cheie gasita anterior (AD respectiv BD).
- Procesul se opreste. Singurele chei ale lui R raman AD si BD.

FORME NORMALE

- Exista cateva seturi de conditii care ne arata ca o schema de relatie este corect proiectata in sensul ca ea nu permite aparitia anomaliilor prezentate la inceputul capitolului.
- Daca schema indeplineste cerintele unui anumit set de conditii se spune ca este in ***forma normala*** asociata acelui set.
- In continuare sunt prezentate formele normale Boyce-Codd si forma normala 3. In finalul acestul capitol va fi prezentata si forma normala 4 care se defineste in functie de alt tip de dependente, si anume dependentele multivalorice.

FNBC - DEFINITIE

Definitie.

- Fie R o schema de relatie si F multimea de dependente functionale asociata.
- Se spune ca R este in forma normala Boyce-Codd (FNBC) daca si numai daca oricare ar fi o dependenta netriviala $X \rightarrow Y$ din F atunci X este supercheie pentru R

FNBC – cont.

- Rezulta ca o schema de relatie este in FNBC daca si numai daca fiecare dependenta din F are in partea stanga o supercheie.
- Nu este necesar ca F sa fie in forma canonica dar nu trebuie sa contina dependente triviale (obtinute din prima axioma - de reflexivitate, de tipul $AB \rightarrow A$ sau $AB \rightarrow AB$)

EXEMPLUL 1

- Relatia R = ABCDE avand
- F = { $A \rightarrow B$, $B \rightarrow A$, $A \rightarrow C$, $D \rightarrow E$ }

- Nu este in forma normala Boyce-Codd deoarece are cheile AD si BD dar nici o dependenta nu are in partea stanga o supercheie a lui R

EXEMPLUL 2

- Relatia Produse = IdP, NumeP, Qty, IdF avand asociata multimea de dependente functionale
- $F_{PRODUSE} = \pi_{PRODUSE}(F) =$ $\{ IdP \rightarrow NumeP, IdP \rightarrow Qty, IdP \rightarrow IdF \}$
- Este in forma normala Boyce-Codd: cheia unica a relatiei este IdP si toate dependentele au in partea stanga o supercheie (asa cum s-a mentionat orice cheie este in acelasi timp si supercheie)

EXEMPLUL 3

- Relatia Produse = IdP, NumeP, Qty, IdF, NumeF, AdresaF avand dependentele:
- F = { IdP→NumeP, IdP→Qty, IdP→ IdF, IdF→NumeF, IdF→AdresaF }
- Nu este in forma normala Boyce-Codd: cheia unica este IdP dar exista dependente care nu au in partea stanga o dupercheie: IdF→NumeF, IdF→AdresaF

FN3 – ATRIBUT PRIM

- Pentru definitia formei normale 3 este necesara definirea notiunii de ***atribut prim***:
- **Definitie.** R o schema de relatie si F multimea de dependente functionale asociata. Un atribut $A \in R$ se numeste atribut prim daca el apartine unei chei a lui R.
- Exemplu: R = ABCDE avand
- $F = \{ A \rightarrow B, B \rightarrow A , A \rightarrow C, D \rightarrow E \}$.
- Cum cheile relatiei sunt AD si BD rezulta ca in R sunt trei atribute prime: A, B si D.

FN3 - DEFINITIE

- **Definitie.** R o schema de relatie si F multimea de dependente functionale asociata. Se spune ca R este in forma normala 3 (FN3) daca si numai daca oricare ar fi o dependenta netriviala

  $X \rightarrow A$ din F atunci

- X este supercheie pentru R

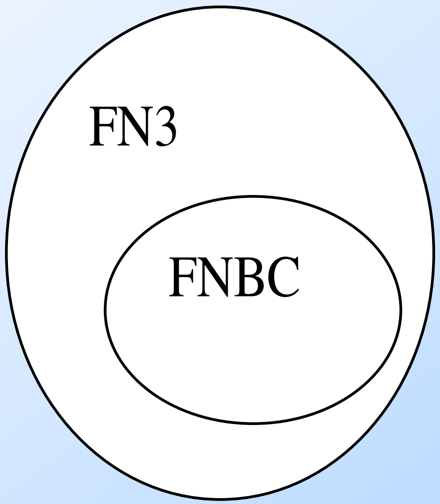
  sau

- A este atribut prim

FN3 – cont.

- Daca in F avem dependente care contin mai multe atribute in partea dreapta putem aplica regula de descompunere pentru a obtine dependente care in partea dreapta au cate un singur atribut.
- Conditia de FNBC este inclusa in definitia FN3. Din acest motiv orice relatie care este in FNBC este implicit si in FN3. Reciproca nu este adevarata.
- De asemenea daca o schema de relatie nu este in FN3 ea nu poate fi nici in FNBC.

FN3 INCLUDE FNBC



EXEMPLUL 1

- Relatia R = ABCD avand
- F = { $AB \rightarrow C$, $AB \rightarrow D$, $D \rightarrow A$ } are cheia unica AB.
- Relatia este in FN3 deoarece primele doua dependente au in partea stanga o supercheie (AB) iar a treia dependenta are in partea dreapta atributul prim A.
- Relatia nu este in FNBC deoarece a treia dependenta violeaza definitia pentru aceasta forma normala (nu are in partea stanga o supercheie.

EXEMPLUL 2

- Relatia R = ABCDE avand
- F = { A $\rightarrow$ B, B $\rightarrow$ A , A $\rightarrow$ C, D $\rightarrow$ E } are cheile AD si BD.
- R nu este in FN3 deoarece dependentele 3 si 4 nu au nici supercheie in partea stanga nici atribut prim in partea dreapta
- R nu e in FNBC deoarece nu e in FN3

EXEMPLUL 3

- Relatia $R = ABCD$ avand
- $F = \{ A \to B, B \to C, C \to D, D \to A \}$ are cheile A, B, C si D. Rezulta ca:
- R este in FNBC deoarece in partea stanga a dependentelor sunt numai superchei
- R este in FN3 deoarece este in FNBC

FN1 si FN2

- Aceste forme normale nu garanteaza eliminarea anomaliilor deci ele nu sunt de dorit pentru schemele de relatie ale unei baze de date a unei aplicatii. Prezentam pe scurt definitia lor.

FN1

- **Definitie:** O relatie R este in **forma normala 1** (FN1) daca pe toate atributele sale exista doar valori atomice ale datelor.
- Semnificatia termenului 'atomic' este similara cu cea de la modelul entitate asociere: valoarea respectiva este intotdeauna folosita ca un intreg si nu se utilizeaza niciodata doar portiuni din aceasta.

FN1 – cont.

- De exemplu, daca intr-o relatie continand date despre persoane avem atributul ***Adresa*** acesta este atomic daca niciodata nu este nevoie sa fie folosite doar anumite portiuni ale sale (strada, numar, etc).

DEP. PARTIALE SI TRANZITIVE

- Fiind data o relatie R si multimea de dependente functionale asociata F putem defini inca doua concepte. Fie A un atribut neprim si X o multime de atribute din R. Atunci:
- **Definitie:** O dependenta functionala $X \rightarrow A$ se numeste ***dependenta partiala*** daca X este strict inclusa intr-o cheie a relatiei R.
- **Definitie:** O dependenta functionala $X \rightarrow A$ se numeste ***dependenta tranzitiva*** daca X nu este inclusa in nici o cheie a relatiei R.

FN2 - DEFINITIE

Definitie:

- Fie R o schema de relatie si F multimea de dependente functionale asociata.
- Se spune ca R este in forma normala 2 (FN2) daca si numai daca F nu contine dependente partiale (dar poate contine dependente tranzitive).

EXEMPLU

- Produse = IdP, NumeP, Qty, IdF, NumeF, AdresaF

- F = { IdP$\rightarrow$NumeP, IdP$\rightarrow$Qty, IdP$\rightarrow$ IdF, IdF$\rightarrow$NumeF, IdF$\rightarrow$AdresaF }

- Este in FN2 pentru ca ultimele doua dependente nu sunt partiale (IdF nu apartine cheii unice IdP).

ATENTIE!

- Asa cum s-a specificat anterior FN2 nu este o forma normala ‘buna’, ea trebuind evitata
- Relatia din exemplul anterior prezinta toate anomaliile enumerate la inceputul acestui capitol.

DESCOMPUNEREA SCHEMELOR DE RELATIE

- Asa cum s-a mentionat anterior, in cazul in care o relatie din baza de date nu este intr-o forma normala buna (FNBC, FN3) pot sa apara diverse anomalii.
- Solutia este inlocuirea relatiei respective cu doua sau mai multe relatii care sa contina aceleasi informatii dar care, fiecare in parte, este in forma normala dorita de proiectant.

DEFINITIE

- Procesul prin care se ‘sparge’ o relatie in mai multe relatii se numeste ***descompunerea unei scheme de relatie***.

Formal putem defini acest concept astfel:

- **Definitie:** Fie R o schema de relatie, R = A1 A2 … Am .

  Se spune ca ρ = (R1, R2, …, Rn) este o ***descompunere*** a lui R daca si numai daca

$$R = R1 \cup R2 \cup \ldots \cup Rn$$

OBSERVATII

- Schemele R1, R2, ..., Rn contin deci atribute din R, fiecare atribut Ai al schemei initiale trebuind sa se regaseasca in cel putin una dintre ele.
- Nu este necesar ca schemele sa fie disjuncte (in practica ele au aproape intotdeauna atribute comune).
- In exemplele de mai jos sunt prezentate cateva descompuneri valide ale unor scheme de relatii (unele insa incorecte din punct de vedere al pastrarii datelor si/sau dependentelor initiale)

EXEMPLUL 1

- Fie relatia R = ABCDE avand
  $F = \{ A \rightarrow B, B \rightarrow A , A \rightarrow C, D \rightarrow E \}$.
- Putem avea descompuneri ca:
  - $\rho 1$ = (ABC, DE),
  - $\rho 2$ = (ABCD, DE),
  - $\rho 3$ = (AB, CD, DE)

EXEMPLUL 2

- Fie relatia Produse = IdP, NumeP, Qty, IdF, NumeF, AdresaF avand dependentele functionale:
- F = { IdP$\rightarrow$NumeP, IdP$\rightarrow$Qty, IdP$\rightarrow$ IdF, IdF$\rightarrow$NumeF, IdF$\rightarrow$AdresaF }

Putem avea o multitudine de descompuneri printre care:

- ρ1 = **(** (IdP, NumeP, Qty, IdF); (NumeF, AdresaF) **)**
- ρ2 = **(** (IdP, NumeP, Qty, IdF); (IdF, NumeF, AdresaF) **)**
- ρ3 = **(** (IdP, NumeP); (Qty, IdF); (NumeF, AdresaF) **)**

SCHEMA si CONTINUT

- Descompunerea actioneaza deci la nivelul ***schemei*** relatiei. Ce se intampla insa cu ***continutul*** acesteia in cazul unei descompuneri?
- Fiecare relatie rezultata va mosteni o parte dintre datele relatiei descompuse si anume proiectia acesteia pe multimea de atribute a relatiei rezultata din descompunere.
- Sa consideram o instanta **r** a relatiei de schema R (instanta unei relatii este o incarcare cu date corecte a acesteia). Atunci instantele pentru relatiile din descompunerea ρ sunt:

$$r_i = \pi_{R_i}(r)$$

EXEMPLU

IdP	NumeP	Qty	IdF	NumeF	AdresaF
101	Imprimanta laser	30	20	Xerox	Str. Daniel Danielopolu 4-6, Sector 1, Bucureşti
105	Calculator PC	20	23	IBM	Bd. D.Cantemir nr.1, Bucuresti
124	Copiator	10	20	Xerox	Str. Daniel Danielopolu 4-6, Sector 1, Bucureşti

ρ1 = **(** (IdP, NumeP, Qty, IdF); (NumeF, AdresaF) **)**

IdP	NumeP	Qty	IdF
101	Imprimanta laser	30	20
105	Calculator PC	20	23
124	Copiator	10	20

NumeF	AdresaF
Xerox	Str. Daniel Danielopolu 4-6, Sector 1, Bucureşti
IBM	Bd. D.Cantemir nr.1, Bucuresti

$\rho 2 = \big((IdP, NumeP, Qty, IdF); (IdF, NumeF, AdresaF) \big)$

IdP	NumeP	Qty	IdF
101	Imprimanta laser	30	20
105	Calculator PC	20	23
124	Copiator	10	20

IdF	NumeF	AdresaF
20	Xerox	Str. Daniel Danielopolu 4-6, Sector 1, Bucureşti
23	IBM	Bd. D.Cantemir nr.1, Bucuresti

ρ3 = ((IdP, NumeP); (Qty, IdF); (NumeF, AdresaF))

IdP	NumeP
101	Imprimanta laser
105	Calculator PC
124	Copiator

Qty	IdF
30	20
20	23
10	20

NumeF	AdresaF
Xerox	Str. Daniel Danielopolu 4-6, Sector 1, Bucureşti
IBM	Bd. D.Cantemir nr.1, Bucuresti

REZULTAT

- Observam din aceste exemple ca in cazul primei si ultimei descompuneri nu putem reconstrui prin join sau alti operatori relationali relatia initiala.
- In cazul in care descompunerea nu s-a facut corect putem pierde:
  - Datele relatiei initiale
  - Dependentele functionale ale relatiei initiale.
- In paragrafele urmatoare sunt prezentati algoritmi prin care putem detecta daca prin descompunere se pierd date sau dependente.

JOIN FARA PIERDERI

- Conditia pentru a nu se pierde date prin descompunere este ca relatia initiala sa poata fi reconstruita exact prin joinul natural al relatiilor rezultate, fara tupluri in minis sau in plus.
- Formal, definitia este urmatoarea:

JOIN FARA PIERDERI (2)

Definitie:

- Fie R o schema de relatie,
- F multimea de dependente functionale asociata si
- o descompunere ρ = (R1, R2, ..., Rn) a lui R.

Se spune ca ρ este o descompunere ***cu join fara pierderi in raport cu F*** (prescurtat j.f.p.) daca si numai daca pentru orice instanta **r** a lui R care satisface dependentele F avem ca:

$$r1 \bowtie r2 \bowtie \ldots \bowtie rn = r \quad \text{unde}$$

$$r_i = \pi_{Ri}(r)$$

JOIN FARA PIERDERI (3)

- In exemplul anterior doar descompunerea

$\rho 2 = (\ (IdP, NumeP, Qty, IdF);\ (IdF, NumeF, AdresaF)\ )$

are aceasta proprietate, in cazul celorlalte, din cauza inexistentei coloanelor comune, joinul natural nu se poate efectua.

JOIN FARA PIERDERI (4)

- Faptul ca o descompunere are sau nu aceasta proprietate se poate testa pornind doar de la
  - lista atributelor relatiei initiale,
  - lista atributelor relatiilor din descompunere si
  - multimea de dependente functionale asociata
- Este deci o proprietate a schemei relatiei si nu a instantelor sale

ALGORITM TESTARE JFP

- **Intrare:** Schema de relatie R = A1 A2 ... Am , multimea de dependente functionale F si o descompunere ρ = (R1, R2, ..., Rn)
- **Iesire:** Verdictul daca ρ are sau nu proprietatea j.f.p.
- **Metoda:**

Se construieste o tabela avand **n** linii si **m** coloane. Liniile sunt etichetate cu elementele lui ρ iar coloanele cu atributele lui R.

Elementul (i,j) al tabelei va fi egal cu

  - a_j daca Aj $\in$ Ri sau
  - b_{ij} altfel.

ALGORITM – cont.

- Se parcurg dependentele $X \rightarrow Y$ din F. Daca doua (sau mai multe) linii din tabela au aceiasi simboli pe coloanele X aceste linii se egaleaza si pe coloanele din Y astfel:
  - Daca pe o coloana din Y apare un aj atunci toate elementele de pe acea coloana din liniile respective devin aj
  - Daca pe o coloana din Y nu apare nici un aj atunci se alege unul dintre elementele de tip bij si toate elementele de pe acea coloana din liniile respective devin egale cu acel bij

ALGORITM – cont.

Procesul se opreste:

- Fie cand s-a obtinut o linie in tabela care contine doar a-uri, caz in care descompunerea ρ **are** proprietatea j.f.p.
- Fie cand la o parcurgere a dependentelor nu mai apar schimbari in tabela si nu s-a obtinut o linie doar cu a-uri. In acest caz descompunerea ρ **nu are** proprietatea j.f.p.

ALGORITM – cont.

- In literatura de specialitate se poate gasi demonstratia faptului ca acest algoritm determina corect daca o descompunere are proprietatea j.f.p. sau nu.

EXEMPLUL 1

- Exemplul 1: Fie R = ABCDE, F = { A → B, A → C, A → D, D → E } si o descompunere a lui R ρ = (ABCD, DE)
- Construim tabelul din algoritm:

	A	B	C	D	E
ABCD	a1	a2	a3	a4	~~b15~~ a5
DE	b21	b22	b23	a4	a5

EXEMPLUL 1 – cont.

- La prima parcurgere, pentru dependentele A $\rightarrow$ B , A $\rightarrow$ C , A $\rightarrow$ D nu gasim doua linii cu aceleasi valori pe coloana A
- Pentru dependenta D $\rightarrow$ E cele doua linii sunt egale pe coloana D (simbolul a4).
- Le egalam si pe coloana E: cum pe aceasta coloana exista a5 rezulta ca b15 devine egal cu a5.
- S-a obtinut o linie numai cu a-uri, deci descompunerea are proprietatea de join fara pierderi.

	A	B	C	D	E
ABCD	a1	a2	a3	a4	~~b15~~ a5
DE	b21	b22	b23	a4	a5

EXEMPLUL 2

- Fie relatia R = ABCDE, ρ = (AB, BC, CDE).
- F = { A → B, AC → D, D → E }
- La prima trecere nu apar modificari in tabel. Procesul se opreste si ρ nu are proprietatea j.f.p.

	A	B	C	D	E
AB	a1	a2	b13	b14	b15
BC	b21	a2	a3	b24	b25
CDE	b31	b32	a3	a4	a5

EXEMPLUL 3

- Fie R = ABCDE, ρ = (BCE, AB, ACD)
- F = { C → E (1), A → C (2), B → D (3), D → E (4), E → B (5) } (dep. numerotate intre paranteze)

	A	B	C	D	E
BCE	b11	a2	a3	b14	a5
AB	a1	a2	~~b23~~ a3 (2)	~~b24~~ b14 (3)	~~b25~~ a5 (4)
ACD	a1	~~b32~~ a2 (5)	a3	a4	~~b35~~ a5 (1)

EXEMPLUL 3 – cont.

- Prima trecere:
- Din C $\rightarrow$ E b35 devine a5
- Din A $\rightarrow$ C b23 devine a3

	A	B	C	D	E
BCE	b11	a2	a3	b14	a5
AB	a1	a2	~~b23~~ a3 (2)	~~b24~~ b14 (3)	~~b25~~ a5 (4)
ACD	a1	~~b32~~ a2 (5)	a3	a4	~~b35~~ a5 (1)

EXEMPLUL 3 – cont.

- Din B $\rightarrow$ D b24 devine b14
- Din D $\rightarrow$ E b25 devine a5
- Din E $\rightarrow$ B b32 devine a2
- linie doar cu a-uri => j.f.p.

	A	B	C	D	E
BCE	b11	a2	a3	b14	a5
AB	a1	a2	~~b23~~ a3 (2)	~~b24~~ b14 (3)	~~b25~~ a5 (4)
ACD	a1	~~b32~~ a2 (5)	a3	a4	~~b35~~ a5 (1)

OBSERVATII

- In exemplele de mai sus a fost suficienta o singura trecere prin dependente.
- Exista insa situatii cand sunt necesare mai multe treceri pana procesul se opreste.
- In cazul in care descompunerea are doar doua elemente se poate testa daca are proprietatea de join fara pierderi si altfel

TEST JFP

- Fie R o schema de relatie,
- F multimea de dependente functionale asociata si
- $\rho = (R1, R2)$ o descompunere a sa.

Atunci ρ are proprietatea de join fara pierderi daca una din dependentele urmatoare se poate deduce din F:

- $(R1 \cap R2) \rightarrow (R1 - R2)$ sau
- $(R1 \cap R2) \rightarrow (R2 - R1)$

TEST JFP – cont.

- Pentru a testa daca dependenta se poate deduce din F este suficient sa calculam inchiderea lui ($R1 \cap R2$).
- Daca ea contine fie pe ($R1 - R2$) fie pe ($R2 - R1$) atunci descompunerea este cu join fara pierderi.

EXEMPLU

- Fie R= ABCDE,
- $F = \{ A \rightarrow B, A \rightarrow C, A \rightarrow D, D \rightarrow E \}$
- ρ = (ABCD, DE).

Avem R1 = ABCD, R2 = DE. Rezulta ca:

- (R1 – R2) = ABCD – DE = ABC
- (R2 – R1) = DE – ABCD = E
- $(R1 \cap R2) = D$

EXEMPLU – cont.

Cele doua dependente sunt:

- $(R1 \cap R2) \rightarrow (R1 - R2)$ devine

  $D \rightarrow ABC$

- $(R1 \cap R2) \rightarrow (R2 - R1)$ devine

  $D \rightarrow E$

Ultima este chiar o dependenta din F deci se poate deduce din F deci ρ are proprietatea de join fara pierderi.

PASTRARE DEPENDENTE

- O a doua problema in cazul descompunerii unei scheme R avand dependentele F in mai multe relatii R1, R2, ..., Rn este aceea a pastrarii corelatiilor intre date, corelatii date de dependentele functionale din F.
- Fiecare relatie R_i va mosteni o multime de dependente data de proiectia multimii de dependente functionale F pe R_i

$$F_i = \pi_{Ri}(F)$$

EXEMPLU

- Fie relatia Produse = IdP, NumeP, Qty, IdF, NumeF, AdresaF avand dependentele functionale:
- F = { $IdP \rightarrow NumeP$, $IdP \rightarrow Qty$, $IdP \rightarrow IdF$, $IdF \rightarrow NumeF$, $IdF \rightarrow AdresaF$ }
- In cazul descompunerii $\rho 2$ = (R1, R2) unde:
  - R1 = (IdP, NumeP, Qty, IdF)
  - R2 = (IdF, NumeF, AdresaF)

cele doua relatii mostenesc urmatoarele dependente:

$F_{R1} = \pi_{R1}(F) = \{ IdP \rightarrow NumeP, IdP \rightarrow Qty, IdP \rightarrow IdF \}$

$F_{R2} = \pi_{R2}(F) = \{ IdF \rightarrow NumeF, IdF \rightarrow AdresaF \}$

PASTRARE DEPENDENTE (2)

- Dupa cum se observa toate dependentele relatiei initiale sunt pastrate fie in FR1 fie in FR2.
- Exista insa si cazuri in care unele dependente din F nu mai pot fi regasite in multimile de dependente asociate schemelor din descompunere si nu se pot deduce din acestea.
- In primul caz se spune ca descompunerea pastreaza dependentele iar in al doilea ca descompunerea nu pastreaza dependentele.

DEFINITIE

- Fie R o schema de relatie, F multimea de dependente functionale asociata, o descompunere ρ = (R1, R2, …, Rn) a lui R si $F_i = \pi_{Ri}(F)$ multimile de dependente functionale ale elementelor descompunerii.
- Se spune ca ρ ***pastreaza dependentele*** din F daca si numai daca orice dependenta din F poate fi dedusa din:

$$\cup_{i=1..n} (Fi).$$

DEFINITIE – cont.

- Rezulta ca o descompunere pastreaza dependentele daca si numai daca:

$$F \subseteq (\cup_{i=1..n} (Fi\ ))^+$$

- Din pacate atat proiectia unei multimi de dependente cat si incluziunea de mai sus implica un calcul de inchidere a unei multimi de dependente (F si respectiv reuniunea multimilor F_i).
- Exista si in acest caz un algoritm pentru a testa daca o dependenta este sau nu pastrata dupa descompunere fara a fi necesar efectiv calculul multimilor Fi

ALGORITM DE TESTARE A PASTRARII DEPENDENTELOR

- **Intrare:** o schema de relatie R, multimea de dependente functionale asociata F si o descompunere $\rho = (R1, R2, \ldots, Rn)$
- **Iesire:** verdictul daca ρ pastreaza sau nu dependentele
- **Metoda:** Pentru fiecare dependenta

$$X \rightarrow Y \text{ din } F$$

se procedeaza astfel:

ALGORITM – cont.

- Se porneste cu o multime de atribute Z

$$Z = X$$

- Se parcurg repetat elementele descompunerii ρ.
- Pentru fiecare R_i se calculeaza o noua valoare a lui Z astfel:
- $Z = Z \cup ((Z \cap R_i)^+ \cap R_i)$

ALGORITM – cont.

- Procesul se opreste in momentul cand Z ramane neschimbat la o parcurgere a elementelor Ri.
- Daca $Y \subseteq Z$ atunci dependenta $X \rightarrow Y$ este pastrata, altfel nu e pastrata

Daca toate dependentele din F sunt pastrate inseamna ca ρ pastreaza dependentele din F.

EXEMPLUL 1

- Fie R = ABCDE,
- ρ = (BCE, AB, ACD)
- F = { $C \rightarrow E$, $A \rightarrow C$, $B \rightarrow D$, $D \rightarrow E$, $E \rightarrow B$ }

Se observa ca dependentele $C \rightarrow E$, $A \rightarrow C$ si $E \rightarrow B$ sunt pastrate: ele apartin proiectiei lui F pe BCE (prima si ultima) si ACD (a doua).

- Raman de testat dependentele $B \rightarrow D$ si $D \rightarrow E$. Sa aplicam algoritmul pentru $B \rightarrow D$:

EXEMPLUL 1 – cont.

- Initial $Z = B$

Trecerea 1 prin elementele lui ρ:

- Pentru BCE:

  $Z = B \cup ((B \cap BCE)^{+} \cap BCE) =$

  $B \cup (BDE \cap BCE) = BE$

- Pentru AB:

  $Z = BE \cup ((BE \cap AB)^{+} \cap AB) =$

  $BE \cup (BDE \cap AB) = BE$

- Pentru ACD:

  $Z = BE \cup ((BE \cap ACD)^{+} \cap AB) = BE \cup \varnothing = BE$

EXEMPLUL 1 – cont.

- La urmatoarea trecere Z ramane neschimbat si procesul se opreste.
- Cum $\{D\} \not\subset BE$ rezulta ca dependenta $B \rightarrow D$ nu este pastrata deci ρ nu pastreaza dependentele

EXEMPLUL 2

- Fie schema de relatie R = ABCD,
- $F = \{ A \to B, A \to C, C \to D, D \to A\}$
- $\rho = (ABC, CD)$.

Trebuie sa testam daca $D \to A$ este pastrata (celelalte dependente se regasesc direct in proiectiile lui F pe elementele descompunerii).

EXEMPLUL 2 – cont.

- Initial $Z = D$

Prima trecere prin elementele lui ρ:

- Pentru ABC:

  $Z = D \cup ((D \cap ABC)^+ \cap ABC) =$

  $D \cup \varnothing = D$

- Pentru CD:

  $Z = D \cup ((D \cap CD)^+ \cap CD) =$

  $D \cup (ABCD \cap CD) = CD$

EXEMPLUL 2 – cont.

- A doua trecere prin elementele lui ρ:
- Pentru ABC:

  $Z = CD \cup ((CD \cap ABC)+ \cap ABC) =$

  $CD \cup (ABCD \cap ABC) = ABCD.$
- Stop. Am obtinut ca $A \subseteq Z$, deci dependenta $D \rightarrow A$ este pastrata, deci ρ pastreaza dependentele.

ALGORITMI DE DESCOMPUNERE

- Algoritmii de testare al pastrarii dependentelor si a joinului fara pierderi pot fi aplicati atunci cand descompunerea unei scheme de relatie se face ‘de mana’, pe baza experientei pe care o are proiectantul bazei de date.
- Exista insa algoritmi simpli care, pornind de la o schema de relatie si multimea de dependente functionale asociata ne duc direct la o descompunere care este in FN3 sau FNBC si in plus au proprietatea de join fara pierderi (deci nu se pierd date prin descompunere) si/sau de pastrare a dependentelor.

FN3 + PASTRARE DEP.

- Fie R o schema de relatie si F multimea de dependente functionale asociata:

$$F = \{ X1 \rightarrow Y1,\ \ X2 \rightarrow Y2,\ \ \ldots\ Xn \rightarrow Yn \}$$

- Atunci descompunerea

$$\rho = (X1Y1,\ \ X2Y2,\ \ \ldots\ XnYn)$$

este o descompunere in FN3 cu pastrarea dependentelor.

(cu X_iY_i am notat $X_i \cup Y_i$)

OBSERVATII

- Toate dependentele sunt pastrate: dependenta $X_i \rightarrow Y_i$ este in proiectia lui F pe X_iY_i
- Pentru a minimiza numarul de elemente din descompunere se aplica regula reuniunii: daca avem mai multe dependente care au aceeasi parte stanga le reunim intr-una singura.
- Daca in descompunere exista doua elemente X_iY_i si X_jY_j astfel incat $X_iY_i \subseteq X_jY_j$ atunci X_iY_i se elimina.
- In literatura de specialitate exista demonstratia faptului ca fiecare schema din descompunere este in FN3.

EXEMPLUL 1

- $R = ABCDE$,
- $F = \{ A \to B, A \to C, A \to D, D \to E \}$.

Rescriem prin reuniune multimea de dependente functionale:

$$F = \{ A \to BCD, D \to E \}.$$

Rezulta din algoritm descompunerea

$$\rho = (ABCD, DE)$$

EXEMPLUL 2

- Produse = IdP, NumeP, Qty, IdF, NumeF, AdresaF avand dependentele functionale:

F = { IdP$\rightarrow$NumeP, IdP$\rightarrow$Qty, IdP$\rightarrow$ IdF, IdF$\rightarrow$NumeF, IdF$\rightarrow$AdresaF }

- Rescriem multimea de dependente. Raman doar doua dependente:

F = { IdP$\rightarrow$NumeP, Qty, IdF; IdF$\rightarrow$NumeF, AdresaF }

- Descompunerea in FN3 cu pastrarea dependentelor va fi:

ρ = ((IdP, NumeP, Qty, IdF), (IdF, NumeF, AdresaF))

FN3 + PASTRARE DEP. +JFP

- Daca la descompunerea obtinuta prin algoritmul anterior adaugam o cheie a relatiei (ca element al descompunerii) vom obtine o descompunere care are atat proprietatea de join fara pierderi cat si pe cea a pastrarii dependentelor.
- Formal putem scrie algoritmul astfel:

FN3+PASTRARE DEP. +JFP(2)

- Fie R o schema re relatie si F multimea de dependente functionale asociata, cu

$F = \{ X1 \rightarrow Y1,\ X2 \rightarrow Y2,\ \ldots Xn \rightarrow Yn \}$

si X o cheie pentru R

- Atunci descompunerea

$$\rho = (X, X1Y1,\ X2Y2,\ \ldots XnYn)$$

este o descompunere in FN3 cu pastrarea dependentelor si join fara pierderi.

FN3+PASTRARE DEP. +JFP(3)

- Pastrarea dependentelor este evidenta, ca mai sus.
- Demonstratia faptului ca descompunerea are si proprietatea de join fara pierderi se gaseste in literatura de specialitate.
- Observatie: Daca vreunul dintre elementele de forma X_iY_i contin deja o cheie a lui R atunci nu este necesara adaugarea unui element suplimentar in descompunere.

EXEMPLUL 1

- Pentru relatiile din exemplele anterioare descompunerea ramane aceeasi deoarece:
- In cazul relatiei R = ABCDE cheia este A, deja inclusa in ABCD, deci descompunerea ramane ρ = (ABCD, DE).
- In cazul relatiei PRODUSE de asemenea cheia este IdP, inclusa deja intr-unul din elementele descompunerii.

EXEMPLUL 2

- Fie R = ABCDE,
- $F = \{ A \rightarrow B, B \rightarrow A, A \rightarrow C, D \rightarrow E \}$.

Cheile relatiei sunt AD si BD.

- Rescriem multimea de dependente:

$$F = \{ A \rightarrow BC, B \rightarrow A, D \rightarrow E \}.$$

- Rezulta descompunerea cu pastrarea dependentelor: ρ = (ABC, AB, DE). Cum AB e inclus in ABC rezulta in final ρ = (ABC, DE).
- Cum elementele descompunerii nu contin vreo cheie a lui R, o adaugam. Obtinem in final descompunerea ρ = (AD, ABC, DE) (sau pt. cealalta cheie BD in loc de AD)

FNBC + JFP

- Fie R o schema de relatie si F multimea de dependente functionale asociata, F in forma canonica:

$$F = \{ X1 \rightarrow A1,\ X2 \rightarrow A2,\ \ldots\ Xn \rightarrow An \}.$$

- Putem calcula descompunerea in FNBC cu join fara pierderi iterativ:

FNBC + JFP (2)

- Initial $\rho = (R)$
- La fiecare pas se alege o schema T care contine o dependenta de forma $X \rightarrow A$ care violeaza conditiile de FNBC.
- Schema respectiva este inlocuita in ρ prin T1 si T2 unde
  - $T1 = XA$
  - $T2 = T - \{A\}$
- Procesul se opreste cand in ρ nu mai exista elemente care nu sunt in FNBC

EXEMPLU

- Fie relatia R = ABCD
- F = { AB $\rightarrow$ C, AB $\rightarrow$ D, D $\rightarrow$ A }.
- Cheia relatiei este AB.

Relatia este in FN3 dar nu este in FNBC din cauza dependentei D $\rightarrow$ A care nu are in partea stanga o supercheie a lui R.

EXEMPLU – cont.

- Initial: ρ = (R) = (ABCD)
- Alegem dependenta D $\rightarrow$ A care violeaza conditia de FNBC.
- Inlocuim T = ABCD cu T1 = DA si T2 = ABCD – A = BCD.
- T1 mosteneste de la T dependenta D $\rightarrow$ A, cheia va fi D si T1 e in FNBC
- T2 mosteneste de la T dependenta DB $\rightarrow$ C. Cheia va fi DB si T2 e in FNBC.
- Rezulta ca descompunerea in FNBC cu join fara pierderi este ρ = (AD, BCD).

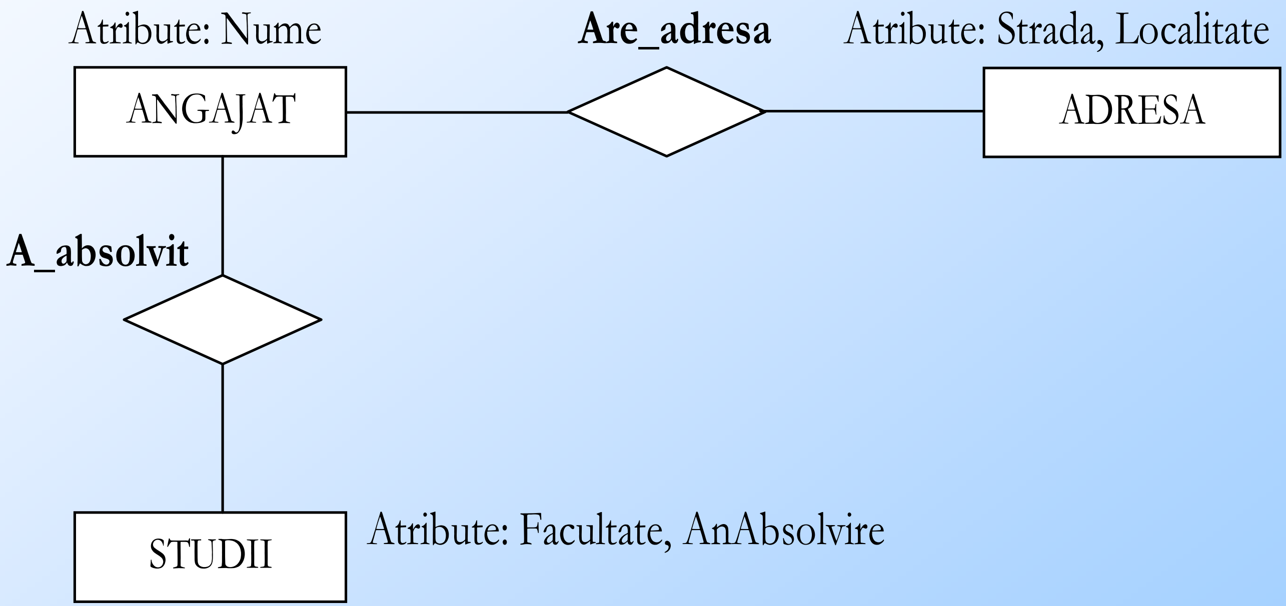
OBSERVATII

- Dependenta mostenita de T2 este din F^+.
- Ea se deduce astfel: Din $D \rightarrow A$ prin augmentare cu B obtinem $DB \rightarrow AB$ si impreuna cu dependenta $AB \rightarrow C$, prin tranzitivitate obtinem $DB \rightarrow C$.
- Analog din $AB \rightarrow D$ se deduce $DB \rightarrow D$ dar aceasta este o dependenta triviala (partea dreapta e inclusa in cea stanga).
- In multe cazuri este nevoie de mai multe iteratii, relatiile de tip T2 (egale in algoritm cu T – A) nefiind uneori in FNBC. Ele se descompun din nou in acelasi fel.

DEPENDENTE MULTIVALORICE

- Exista situatii in care, desi o relatie este in forma normala Boyce Codd, instantele sale contin date redundante.
- Acest fapt se datoreaza unei proiectari defectuoase in care in aceeasi relatie sunt stocate date care apartin mai multor entitati si a cel putin doua asocieri multi-multi.

EXEMPLU - Diagrama



EXEMPLU – cont.

- Ambele asocieri sunt multi-multi: un angajat poate sa fie absolvent al mai multor facultati si in acelasi timp poate avea mai multe adrese (de exemplu una pentru domiciliul stabil si alta pentru rezidenta temporara la un moment dat).
- In cazul in care toate datele din aceasta diagrama sunt stocate intr-o singura tabela putem avea urmatoarea incarcare cu date corecte:

EXEMPLU - Date

Nume	Strada	Localitate	Facultate	AnAbsolvire
Vasile	Viitorului	Ploiesti	Automatica	2000
Vasile	Viitorului	Ploiesti	Comert	2004
Vasile	Dreapta	Bucuresti	Automatica	2000
Vasile	Dreapta	Bucuresti	Comert	2004
Mariana	Revolutiei	Timisoara	Constructii	1998
Mariana	Revolutiei	Timisoara	Drept	2003
Mariana	Revolutiei	Timisoara	Master ASE	2006
Mariana	Calea Vitan	Bucuresti	Constructii	1998
Mariana	Calea Vitan	Bucuresti	Drept	2003
Mariana	Calea Vitan	Bucuresti	Master ASE	2006

EXEMPLU – cont.

- Putem observa ca nu exista nici o dependenta functionala netriviala valida pentru aceasta relatie, deci nu exista dependente care sa violeze conditiile FNBC.
- Ca urmare relatia este in FNBC avand ca singura cheie posibila multimea tuturor atributelor relatiei: din axioma de reflexivitate (A1) putem obtine dependenta:

Nume,Strada,Localitate,Facultate,AnAbsolvire $\to$ Nume,Strada,Localitate,Facultate,AnAbsolvire

OBSERVAM CA:

- Desi relatia este in FNBC adresa si facultatea absolvita de un angajat sunt prezente repetat in relatie: adresa pentru fiecare facultate absolvita iar facultatea pentru fiecare adresa a angajatului.
- Exemplul de mai sus sugereaza faptul ca seturile de atribute {Strada, Localitate} si {Facultate, AnAbsolvire} sunt independente unele de altele, in sensul ca fiecare adresa apare cu fiecare facultate absolvita de un angajat si reciproc.
- Astfel de situatii sunt modelate cu un nou tip de dependente numite ***dependente multivalorice (DMV)***.

DEFINITIE DMV

Definitie: Fie o relatie R si doua multimi de atribute X si Y incluse in R.

- Se spune ca X multidetermina Y sau ca exista dependenta multivalorica $X \rightarrow\rightarrow Y$ daca si numai daca ori de cate ori avem doua tupluri ale relatiei t1 si t2 cu t1[X] = t2[X] atunci exista in relatie un tuplu t3 pentru care:
  - t3[X] = t1[X] = t2[X]
  - t3[Y] = t1[Y] si t3[R-X-Y] = t2[R-X-Y]

VIZUALIZARE

	X	Y	R-X-Y
t1	AAA	BBB	CCC
t2	AAA	DDD	EEE
t3	AAA	BBB	EEE

CONSECINTA

- O consecinta interesanta a acestei definitii este ca, daca inversam tuplurile t1 si t2, rezulta ca exista si un tuplu t4 pentru care
  - t4[X] = t1[X] = t2[X]
  - t4[Y] = t2[Y] si t4[R-X-Y] = t1[R-X-Y]

ALTA CONSECINTA

- Tot din aceasta definitie rezulta ca daca in R exista dependenta multivalorica

$$X \rightarrow\rightarrow Y$$

atunci exista si dependenta

$$X \rightarrow\rightarrow R - X - Y$$

Acest fapt va fi prezentat in paragraful urmator ca axioma de complementare a dependentelor multivalorice.

EXEMPLU

- Intorcandu-ne la exemplul anterior rezulta ca in relatia continand date despre angajati, studii si adrese avem urmatoarele dependentele multivalorice (a doua fiind obtinuta din prima prin complementare):
  - Nume $\rightarrow\rightarrow$ Strada, Localitate
  - Nume $\rightarrow\rightarrow$ Facultate, AnAbsolvire
- Intradevar, daca luam in considerare pentru t1 si t2 tuplurile 2 si 3 din relatie:

EXEMPLU – cont.

Nume	Strada	Localitate	Facultate	AnAbsolvire
Vasile	Viitorului	Ploiesti	Comert	2004
Vasile	Dreapta	Bucuresti	Automatica	2000

gasim in relatie pe prima pozitie si tuplul t3 de forma:

Nume	Strada	Localitate	Facultate	AnAbsolvire
Vasile	Viitorului	Ploiesti	Automatica	2000

In acelasi timp gasim pe pozitia 4 si tuplul t4:

Nume	Strada	Localitate	Facultate	AnAbsolvire
Vasile	Dreapta	Bucuresti	Comert	2004

ALTA ALEGERE t3 SI t4

Nume	Strada	Localitate	Facultate	AnAbsolvire
Vasile	Viitorului	Ploiesti	Automatica	2000
Vasile	Viitorului	Ploiesti	Comert	2004

t3:

Nume	Strada	Localitate	Facultate	AnAbsolvire
Vasile	Viitorului	Ploiesti	Comert	2004

t4:

Nume	Strada	Localitate	Facultate	AnAbsolvire
Vasile	Viitorului	Ploiesti	Automatica	2000

CE REMARCAM?

- Observam ca t3 = t2 si t4 = t1 ceea ce este corect pentru ca in definitia dependentelor multivalorice nu se cere ca t3 sa fie diferit de t1 si t2.
- **Consecinta importanta:** orice dependenta functionala este in acelasi timp si o dependenta multivalorica:
- Fie relatia R si o dependenta functionala $X \rightarrow Y$ pentru R.
- Atunci daca doua tupluri t1 si t2 au aceleasi valori pe atributele X vor avea aceleasi valori si pe atributele Y. Rezulta ca t2 indeplineste conditiile pentru t3 din definitia dependentelor multivalorice:

$X \rightarrow Y$

	X	Y	R-X-Y
t1	AAA	BBB	CCC
t2	AAA	BBB	DDD
t3 este t2	AAA	BBB	DDD

EXEMPLU

- Exemplu: Fie relatia Produse anterioara.
- In aceasta avem dependenta functionala:

$$IdF \rightarrow NumeF, AdresaF$$

- Avem doua tupluri cu aceleasi valori pe IdF:

EXEMPLU – cont.

	IdP	NumeP	Qty	IdF	NumeF	AdresaF
t1	101	Imprimanta laser	30	20	Xerox	Str. Daniel Danielopolu 4-6, Sector 1, Bucureşti
t2	124	Copiator	10	20	Xerox	Str. Daniel Danielopolu 4-6, Sector 1, Bucureşti

EXEMPLU – cont.

- In acest caz putem forma tuplul t3 astfel:
  - Pe IdF valoarea 20
  - Pe NumeF si adresaF valorile din primul tuplu
  - Pe restul atributelor valorile din al doilea tuplu.
- Obtinem t3 identin cu t2:

EXEMPLU – cont.

	IdP	NumeP	Qty	IdF	NumeF	AdresaF
t3	124	Copiator	10	20	Xerox	Str. Daniel Danielopolu 4-6, Sector 1, Bucureşti

AXIOME – A4. COMPLEMENTARE

- Urmatoarele axiome sunt specifice DMV. Le numerotam incepand cu A4 deoarece intr-o schema de relatie pot fi atat dependente functionale (carora li se aplica axiomele A1-A3 descrire anterior) cat si dependente multivalorice.
- **A4. Complementare:** Fie R o schema de relatie si X, Y $\subseteq$ R.

  Daca $X \rightarrow\rightarrow Y$ atunci si $X \rightarrow\rightarrow (R - X - Y)$

A5 – AUGMENTARE DMV

- **Augmentare pentru DMV:** Fie R o schema de relatie si X, Y, Z, W $\subseteq$ R.
- Daca X $\rightarrow\rightarrow$ Y si Z $\subseteq$ W atunci

$$XW \rightarrow\rightarrow YZ$$

A6 – TRANZITIVITATE DMV

- **A6. Tranzitivitate pentru DMV:** Fie R o schema de relatie si X, Y, Z $\subseteq$ R.
- Daca $X \rightarrow\rightarrow Y$ si $Y \rightarrow\rightarrow Z$ atunci

$$X \rightarrow\rightarrow (Z - Y)$$

A7

Ultimele doua axiome leaga dependentele multivalorice cu cele functionale:

- **A7.** Fie R o schema de relatie si
- X, Y $\subseteq$ R.
- Daca $X \rightarrow Y$ atunci si $X \rightarrow\rightarrow Y$

A8

- **A8.** Fie R o schema de relatie si X, Y, Z, $W \subseteq R$. cu $W \cap Y = \emptyset$
- Daca $X \rightarrow\rightarrow Y$, $Z \subseteq Y$, $W \rightarrow Z$ atunci

$$X \rightarrow Z$$

OBSERVATIE

- Orice dependenta functionala este in acelasi timp si o dependenta multivalorica insa reciproca nu este adevarata: exista dependente multivalorice pentru care in schema relatiei nu avem o dependenta functionala corespunzatoare.
- Exemplu pentru acest fapt este dependenta multivalorica existenta in tabela de angajati din paragraful anterior:

OBSERVATIE – cont.

Nume $\rightarrow\rightarrow$ Strada, Localitate

- In relatie nu exista insa si o dependenta functionala echivalenta de tipul:

Nume $\rightarrow$ Strada, Localitate

Rezulta ca:

- Putem folosi si axiomele A1-A3 dar doar pentru dependente multivalorice care sunt in acelasi timp si dependente functionale.
- Pentru restul dependentelor multivalorice putem folosi doar A4-A6.

REGULI PENTRU DMV

- Exista de asemenea o serie de reguli care se pot deduce din axiome. Toate considera existenta unei scheme de relatie R iar X, Y, Z, W sunt submultimi ale lui R:
- **R1. Reuniune:**

  Daca $X \to\to Y$ si $X \to\to Z$ atunci

  $$X \to\to YZ$$

R2, R3

- **R2. Pseudotranzitivitate**:

Daca $X \to\to Y$ si $WY \to\to Z$ atunci

$$WX \to\to Z - WY$$

- **R3. Pseudotranzitivitate mixta:**

Daca $X \to\to Y$ si $XY \to\to Z$ atunci

$$X \to\to Z - Y$$

R4, R5

- **R4. Diferenta:**

Daca $X \rightarrow\rightarrow Y$ si $X \rightarrow\rightarrow Z$ atunci:

$$X \rightarrow\rightarrow Y - Z$$

$$X \rightarrow\rightarrow Z - Y$$

- **R5. Intersectie:**

Daca $X \rightarrow\rightarrow Y$ si $X \rightarrow\rightarrow Z$ atunci:

$$X \rightarrow\rightarrow Y \cap Z$$

R6, R7 SI R8

- **R6. Eliminare atribute comune:**

Daca $X \rightarrow\rightarrow Y$ atunci:

$$X \rightarrow\rightarrow Y - X$$

- **R7. Toate atributele:**

Daca $X \cup Y = R$ atunci

$$X \rightarrow\rightarrow Y \text{ si } Y \rightarrow\rightarrow X$$

- **R8. Reflexivitate:**

Daca $Y \subseteq X$ atunci $X \rightarrow\rightarrow Y$

INCHIDERE

- Aceste axiome si reguli se pot folosi pentru calculul inchiderii unei multimi de dependente functionale si multivalorice.
- Definitia inchiderii este aceeasi ca la dependentele functionale:
- **Definitie:** Fie R o schema de relatie si G multimea de dependente functionale si multivalorice asociata. Atunci ***inchiderea multimii de dependente*** G, notata G^+, este o multime de dependente (DF si DMV) care sunt in G sau se pot deduce din G folosind axiomele si regulile.

PROIECTIE

- Analog cu cazul dependentelor functionale se poate defini si proiectia unei multimi de dependente functionale si multivalorice pe o multime de atribute:
- **Definitie.** Fie o relatie R, o multime asociata de dependente functionale si multivalorice G si o submultime de atribute $S \subseteq R$. ***Proiectia multimii de dependente G pe S***, notata cu $\pi_S(G)$ este multimea dependentelor din G+ care au si partea stanga si pe cea dreapta incluse in S.

DEPENDENTE MOSTENITE

- In momentul in care o schema de relatie se descompune in doua sau mai multe subscheme, fiecare subschema va mosteni o multime de dependente functionale si multivalorice obtinuta prin proiectia multimii initiale G pe atributele din subschema respectiva.

FORMA NORMALA 4

- Pentru a preintampina redundantele prezentate la inceputul paragrafului 4.5. este bine ca schemele de relatie sa fie intr-o forma normala superioara FNBC.
- Aceasta forma care considera si dependentele multivalorice se numeste **forma normala 4** (FN4).
- Definitia ei este similara cu cea pentru FNBC dar conditia se pune pentru dependentele multivalorice ale relatiei respective:

FN4 - DEFINITIE

- **Definitie:** O schema de relatie R este in forma normala 4 daca orice dependenta multivalorica netriviala $X \rightarrow\rightarrow Y$ are in partea stanga o supercheie

DMV TRIVIALE

- Dependentele multivalorice triviale sunt de doua feluri:
- Dependente provenite din R8, deci cele in care partea dreapta este inclusa in partea stanga: $X \rightarrow\rightarrow Y$ unde $Y \subseteq X$
- Dependente provenite din regula R7:

  $X \rightarrow\rightarrow Y$ pentru $X \cup Y = R$
- Conditia de FN4 spune deci ca orice DMV care nu intra in una din categoriile de mai sus are in partea stanga o supercheie.

EXEMPLU

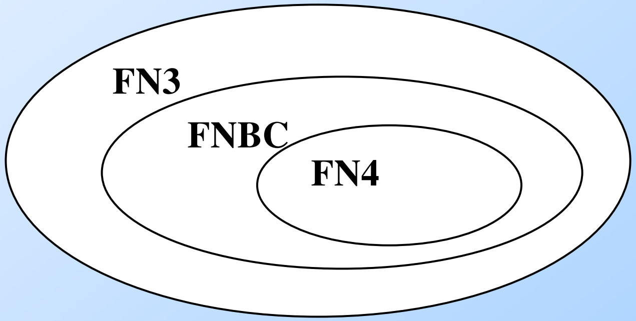
- Relatia angajati-studii-adrese are dependenta netriviala

$$Nume \rightarrow\rightarrow Strada, Localitate$$

- Cum cheia relatiei e multimea tuturor atributelor acesteia, rezulta ca relatia nu este in FN4 deoarece {Nume} nu e supercheie.

RELATIA DINTRE FN

- Relatia dintre formele normal2 FN3, FNBC si FN4 este una de includere, in aceasta ordine.
- Orice relatie in FN4 este in acelasi timp si in FNBC si FN3:



DESCOMPUNERE IN FN4

- Acest algoritm este similar cu cel de descompunere in FNBC dar ia in considerare dependentele multivalorice care violeaza FN4.
- Atentie: dependentele multivalorice ale unei relatii sunt atat cele care provin prin axioma A7 din dependente functionale cat si dependente multivalorice care nu au corespondent in multimea celor functionale.

ALGORITM

- Fie R o schema de relatie si G multimea de dependente multivalorice asociata (consideram ca din G au fost eliminate dependentele triviale). Putem calcula descompunerea in FN4 iterativ:
- Initial $\rho = (R)$
- La fiecare pas se alege o schema T care contine o dependenta de forma $X \rightarrow\rightarrow Y$ care violeaza conditia pentru FN4. Schema respectiva este inlocuita in ρ prin T1 si T2 unde $T1 = XY$ si $T2 = T - Y$
- Procesul se opreste cand in ρ nu mai exista elemente care nu sunt in FN4

EXEMPLU

- Pentru relatia Angajati care nu era in FN4
- Initial

ρ = ((Nume,Strada,Localitate,Facultate,AnAbsolvire))

- Alegem dependenta

Nume $\rightarrow\rightarrow$ Strada, Localitate

care violeaza conditia pentru FN4. Obtinem

- T1 = Nume, Strada, Localitate si
- T2 = Nume, Facultate, AnAbsolvire

EXEMPLU – cont.

- Obtinem ρ = ((Nume, Strada, Localitate), (Nume, Facultate, AnAbsolvire)). Fiecare subschema mosteneste cate o dependenta multivalorica:
  - T1: Nume $\rightarrow\rightarrow$ Strada, Localitate
  - T2: Nume $\rightarrow\rightarrow$ Facultate, AnAbsolvire
- Cum cele doua dependente mostenite de T1 si T2 sunt triviale (contin toate atributele relatiei) rezulta ca cele doua relatii sunt in FN4 deoarece nu exista dependente netriviale care violeaza FN4. Procesul s-a incheiat.

OBSERVATIE

- Fiecare subschema T_i obtinuta la descompunere mosteneste de la relatia originala T proiectia multimii de dependente a lui T (DF si DMV) pe T_i.
- Cum relatia initiala avea doar doua DMV si nici o DF, multimile de dependente pentru T1 si T2 sunt cele din slide-ul anterior.

Sfarsitul capitolului 4:

Proiectarea bazelor de date relationale