

# Baze de date

---

Universitatea “Transilvania” din Brasov

Lect.dr. Costel Aldea  
costel.aldea@gmail.com

# Arhitectura bazei de date cu 3 nivele

---

- Asigurarea independenței fizice și logice a datelor impune adoptarea unei arhitecturi organizată pe cel puțin 3 nivele
- Arhitectura ANSI/SPARC este adoptată de majoritatea sistemelor moderne de baze de date. Ea este organizată astfel:
  - **nivelul intern** (baza de date fizică) – se referă la modul fizic în care sunt stocate datele
  - **nivelul conceptual** – este cel mai apropiat de utilizatori și se referă la modul în care aceștia pot vedea datele
  - **nivelul extern** – este nivelul de interfață dintre celelalte două

# Arhitectura bazei de date cu 3 nivele

## *Exemplu*

---

- Avem
  - o vedere conceptuală
  - vederea internă corespunzătoare
  - două vederi externe, una pentru un utilizator PL/I și alta pentru un utilizator COBOL

# Arhitectura bazei de date cu 3 nivele

## *Exemplu*

---

- La nivel conceptual, baza de date conține informații referitoare la o entitate numită EMPLOYEE. Fiecare apariție din EMPLOYEE are
  - EMPLOYEE\_NUMBER (6 caractere)
  - DEPARTMENT\_NUMBER (4 caractere)
  - SALARY (10 cifre)
- La nivel intern, angajatii sunt reprezentați printr-o înregistrare numită STORED\_EMP (20 de octeți)
  - Prefix care conține informație de control – 6 octeți
  - 3 câmpuri de date corespunzătoare celor trei proprietăți ale angajaților

# Arhitectura bazei de date cu 3 nivele

## *Exemplu*

---

### □ Nivelul extern 1 (PL/I)

```
DCL    1  EMPP ,  
        2  EMP#  CHAR(6)  
        2  SAL  FIXED BIN(31) ;
```

### □ Nivelul extern 2 (COBOL)

```
01  EMPC .  
    02  EMPNO  PIC X(6) .  
    02  DEPTNO PIC X(4) .
```

# Arhitectura bazei de date cu 3 nivele

## *Exemplu*

---

### ❑ Nivelul conceptual

#### **EMPLOYEE**

<b>EMPLOYEE_NUMBER</b>	<b>CHARACTER (6)</b>
<b>DEPARTMENT_NUMBER</b>	<b>CHARACTER (4)</b>
<b>SALARY</b>	<b>NUMERIC (5)</b>

### ❑ Nivelul intern

<b>STORED_EMP</b>	<b>LENGTH=20</b>
<b>PREFIX</b>	<b>TYPE=BYTE (6) , OFFSET=0</b>
<b>EMP#</b>	<b>TYPE=BYTE (6) , OFFSET=6 , INDEX=EMPX</b>
<b>DEPT#</b>	<b>TYPE=BYTE (4) , OFFSET=12</b>
<b>PAY</b>	<b>TYPE=FULLWORD , OFFSET=16</b>

# Arhitectura bazei de date cu 3 nivele

## *Exemplu*

---

- ❑ Utilizatorul PL/I are o vedere externă a bazei de date în care fiecare angajat este reprezentat printr-o înregistrare PL/I care conține două câmpuri
- ❑ Numărul departamentului nu prezintă interes pentru acest utilizator și a fost omis din vedere
- ❑ Tipul de înregistrare a fost definit printr-o structură obișnuită PL/I, conform regulilor acestui limbaj

# Arhitectura bazei de date cu 3 nivele

## *Exemplu*

---

- ❑ Utilizatorul COBOL are o vedere externă în care fiecare angajat este reprezentat printr-o înregistrare COBOL care conține tot două câmpuri. De această data a fost omis salariul
- ❑ Înregistrarea COBOL a fost definită printr-o descriere tipică, conform regulilor acestui limbaj
- ❑ Corespondența între câmpurile de pe cele 3 nivele se face prin *mapare*, o operație realizată de sistemele relaționale



# Arhitectura bazei de date cu 3 nivele

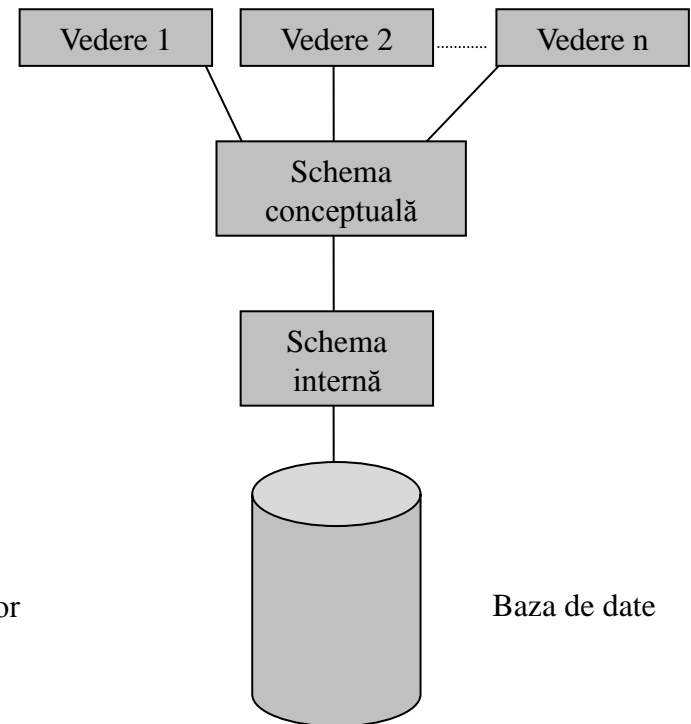
- **Obiectivul** arhitecturii cu 3 nivele este **separarea vederii fiecărui utilizator** asupra bazei de date de modul în care este ea reprezentată fizic

Nivelul extern

Nivelul conceptual

Nivelul intern

Organizarea fizică a datelor



Baza de date

# Arhitectura bazei de date cu 3 nivele

## *Nivelul intern*

---

- Este reprezentarea fizică a bazei de date pe calculator
- Descrîe *CUM* sunt stocate datele în baza de date
- Nivelul intern (baza de date fizică) este o *colecție de fișiere* conținând *datele fizice* la care se adaugă diverse *structuri auxiliare* menite să asigure accesul operativ la date

# Arhitectura bazei de date cu 3 nivele

## *Nivelul intern*

---

- Structurile auxiliare pot fi
  - directoare
  - indecși
  - pointeri
  - tabele de dispersie
- Modul de organizare a bazei de date fizice este în mare măsură influențat de
  - *configurația echipamentelor hardware* care suportă baza de date
  - *sistemul de operare*

# Arhitectura bazei de date cu 3 nivele

## *Nivelul intern*

---

- ❑ Schimbarea sistemului de operare sau modificări în configurația hardware pot atrage modificări ale bazei de date fizice
- ❑ Dacă este satisfăcută condiția de **independență fizică**, aceste modificări în nivelul intern al bazei de date nu vor afecta nivelele superioare ale acesteia

# Arhitectura bazei de date cu 3 nivele

## *Nivelul intern*

---

- Nivelul intern tratează chestiuni cum ar fi:
  - alocarea spațiului de stocare pentru date și indecși
  - descrierea înregistrărilor pentru stocare (cu dimensiunile de stocare pentru date)
  - plasarea înregistrărilor
  - tehnici de comprimare a datelor și de codificare a acestora

# Arhitectura bazei de date cu 3 nivele

## *Nivelul intern*

---

- *Vederea internă* este o reprezentare la nivelul de jos a întregii baze de date
- Constă dintr-o colecție de *înregistrări interne* (*internal records*)
- Vederea internă este diferită de nivelul fizic deoarece nu lucrează în mod direct cu *blocuri* sau *pagini* – unitățile de intrare / ieșire care, în mod obișnuit, au 1K, 2K sau 4K

# Arhitectura bazei de date cu 3 nivele

## *Nivelul intern*

---

- Vederea internă este descrisă de schema internă care
  - Definește diferitele modalități de stocare a datelor
  - Specifică indecșii, reprezentarea câmpurilor, care este secvența fizică din fiecare înregistrare etc.

# Arhitectura bazei de date cu 3 nivele

## *Nivelul intern*

---

- În anumite situații excepționale, programelor aplicație – în particular aplicațiilor de natură utilitară – li se permite să opereze direct la nivelul intern în loc de nivelul extern din considerente de performanță
- Această practică nu este recomandată pentru că
  - prezintă riscuri de securitate
  - prezintă riscuri de integritate
  - programul este dependent de date



# Arhitectura bazei de date cu 3 nivele

## *Nivelul conceptual*

---

- Este o vedere generală a bazei de date
- Acest nivel descrie **CE** date sunt stocate în bază de date și **RELAȚIILE** dintre acestea
- Conține **structura logică** a bazei de date, așa cum este ea văzută de administratorul bazei de date
- Fiecare bază de date are un model conceptual propriu prin care sunt numite și descrise toate entitățile logice din baza de date împreună cu legăturile dintre acestea

# Arhitectura bazei de date cu 3 nivele

## *Nivelul conceptual*

---

- *Vederea conceptuală* este o reprezentare a întregului conținut al bazei de date într-o formă care este abstractă în comparație cu modul fizic de stocare a datelor
- Este, de asemenea, diferită de modul în care datele sunt vizualizate de un anumit utilizator
- *Vederea conceptuală* se vrea a fi o vedere a datelor “așa cum sunt ele în realitate”, în comparație cu modalitățile de vizualizare la care sunt constrânși utilizatorii

# Arhitectura bazei de date cu 3 nivele

## *Nivelul conceptual*

---

- Vederea conceptuală constă dintr-o colecție de *înregistrări conceptuale* (*conceptual records*). Reprezintă o imagine completă a cerințelor organizației privind datele

### Exemple

În descrierea bazei de date a unei societăți comerciale pot apărea concepte ca: angajat, produse, furnizor, beneficiar etc.

# Arhitectura bazei de date cu 3 nivele

## *Nivelul conceptual*

---

- Modelul conceptual *integrează vederile tuturor utilizatorilor* asupra bazei de date, fiind rezultatul unui compromis între cerințele diferiților utilizatori
- Nivelul conceptual reprezintă:
  - toate entitățile, atributele și relațiile dintre ele
  - constrângeri asupra datelor
  - informații semantice asupra datelor
  - informații privind securitatea și integritatea

# Arhitectura bazei de date cu 3 nivele

## *Nivelul conceptual*

---

- Vederea conceptuală este definită prin intermediul *schemei conceptuale*
- Aceasta include definițiile diverselor tipuri de înregistrare conceptuale
- Vederea conceptuală este o vedere a întregului conținut al bazei de date, iar schema conceptuală este o definiție a acestei vederi
- Definițiile din schema conceptuală includ un număr mare de caracteristici adiționale, cum sunt regulile de securitate și integritate

# Arhitectura bazei de date cu 3 nivele

## *Nivelul extern*

---

- Reprezintă *vederea utilizatorului* asupra bază de date
- Acest nivel descrie acea parte a bazei de date care este relevantă pentru fiecare utilizator
- Este ceea ce vede acesta din baza de date, sau modul cum vede acesta baza de date
- Modelul extern este derivat din cel conceptual dar poate avea deosebiri substanțiale față de acesta

# Arhitectura bazei de date cu 3 nivele

## *Nivelul extern*

---

- Prin vederile externe utilizatorii au acces doar la părți bine definite din baza de date, fiindu-le ascunse părțile care nu îi interesează
- Prin modelul extern se realizează **independența logică** a datelor
- Fiecărei vederi îi corespunde o descriere în termenii entităților logice din modelul conceptual

# Arhitectura bazei de date cu 3 nivele

## *Nivelul extern*

---

- Diferite vederi pot avea reprezentări diferite ale acelorași date

### Exemplu

Un utilizator poate vedea datele calendaristice în format an-lună-zi, altul le poate vedea ca zi-lună-an

- Vederile pot include chiar date combinate sau derivate din entități diferite



# Limbajele bazelor de date

---

- Limbajele bazelor de date sunt împărțite în 2 categorii
  - limbaje de definire a datelor (**DDL**)
  - limbaje de manipulare a datelor (**DML**)
- DDL este utilizat pentru a specifica schema bazei de date
  - Exemplu - instrucțiunile DECLARE scrise în PL/I sunt de tip DDL
- DML este utilizat pentru citirea și reactualizarea bazei de date
  - Exemplu - instrucțiunile executabile scrise în PL/I, care transferă date din baza de date sunt de tip DML

# Limbajele bazelor de date

---

- Aceste limbaje sunt numite **sublimbaje de date** deoarece ele nu includ construcții pentru toate necesitățile de calcul, cum sunt cele asigurate de limbajele de nivel înalt
- Multe SGBD au o facilitate de încorporare a sublimbajului într-un limbaj de programare de nivel înalt, cum sunt COBOL, Pascal, C, etc. În acest caz, limbajul de nivel înalt se numește *limbaj gazdă*
- Pentru a compila fișierul încorporat, mai întâi comenzile specifice sublimbajului de date sunt înlocuite prin apeluri de funcții
- Apoi fișierul preprocesat este compilat și rezultatul este plasat într-un modul obiect, legat la o bibliotecă de funcții înlocuite

# Limbajele bazelor de date

## *Data Definition Language (DDL)*

---

- Este un *limbaj descriptiv*
- Permite administratorului bazei de date sau utilizatorului să descrie și să denumească
  - entitățile cerute de aplicație
  - relațiile care pot exista între diferitele entități
- Rezultatul compilării instrucțiunilor DDL este *catalogul de sistem*. Acesta conține *metadatele* – adică datele care descriu obiectele din baza de date

# Limbajele bazelor de date

## *Data Manipulation Language (DML)*

---

- Asigură un set de procedee ce permit operații de bază pentru manipularea datelor din bază de date:
  - inserarea de date noi
  - modificări de date
  - regăsirea datelor
  - ștergerea de date
- Limbajele DML pot fi de două tipuri:
  - **procedurale** - specifică modul *cum* trebuie să fie obținut rezultatul unei instrucțiuni DML
  - **neprocedurale** descriu numai *ce* rezultat trebuie obținut – limbajul SQL

# Modele de date și modelarea conceptuală

---

- Modelul de date
  - Este o colecție integrată de concepte necesare
    - *descrierii* datelor
    - *relațiilor* dintre date
    - *constrângerilor* impuse datelor

# Modelul de date

---

- Un model de date are următoarele trei componente:
  - o parte **structurală**
    - constă dintr-un set de reguli conform cărora sunt construite bazele de date
  - o parte de **manipulare**
    - definește tipurile de operații care sunt permise asupra datelor
  - un set de **reguli de integritate**
    - garantează că datele sunt corecte

# Modelul de date

---

- **Scopul** unui model este să reprezinte datele și să le facă înțelese
- Pentru arhitectura ANSI - SPARC a bazei de date, se pot identifica trei modele de date:
  - un model de date **extern**
    - pentru a reprezenta vederea fiecărui utilizator
  - un model de date **conceptual**
    - pentru a reprezenta vederea logică, generală, care este independentă de SGBD
  - un model de date **intern**
    - pentru a reprezenta schema conceptuală, în așa fel încât să poată fi înțeleasă de SGBD

# Modelul de date

---

- Modelele de date se pot clasifica în trei categorii principale:
  - modele de date **bazate pe obiecte**
  - modele de date **bazate pe înregistrări**
  - modele de date **fizice**



# Modelul de date

---

- Modelele de date **fizice** descriu **cum** sunt stocate datele pe calculator
- Reprezintă informații despre
  - structura înregistrărilor
  - ordinea înregistrărilor
  - căile de acces
- Nu există la fel de multe modele de date fizice ca cele logice, motiv pentru care vom prezenta mai amănunțit numai primele două categorii de modele de date

# Modele de date bazate pe obiecte

---

- În modelele de date **bazate pe obiecte** se utilizează concepte ca
  - entitate
  - atribut
  - relație
- O **entitate** este un obiect distinct (persoană, loc, lucru, concept, eveniment) care va fi reprezentat în baza de date
- Un **atribut** este o proprietate care descrie un anumit aspect al obiectului pe care dorim să-l înregistrăm
- O **relație** este o asociere între entități

# Modele de date bazate pe obiecte

---

## □ Exemple

### ■ Entitate/Relație (Semantic)

- constituie una din tehnicile principale de proiectare conceptuală a bazelor de date

### ■ Orientat spre obiecte

- extinde definiția unei entități, pentru a include
  - *atributele* care descriu *starea* obiectului
  - *acțiunile* acestuia, respectiv *comportamentul*
- se spune că obiectul **încapsulează** atât starea cât și comportamentul

## Modele de date bazate pe înregistrări

---

- ❑ Într-un astfel de model, baza de date constă dintr-un număr de înregistrări cu format fix, posibil de tipuri diferite
- ❑ Fiecare tip de înregistrare definește un număr fix de câmpuri, fiecare având o lungime fixă

# Modele de date bazate pe înregistrări

---

- Există 3 tipuri principale de modele de date bazate pe înregistrări
  - relațional
  - în rețea
  - ierarhic
- Ultimele două au fost realizate cu aproape 10 ani înaintea celui relațional, așa încât legăturile lor cu conceptele tradiționale de prelucrare a fișierelor sunt mai evidente

# Modelul de date relațional

---

- Se bazează pe conceptul de **relații matematice**
- Datele și relațiile sunt reprezentate sub formă de **tabele**, fiecare având un număr de coloane cu o denumire unică

# Modelul de date relațional

---

- ❑ Modelul de date relațional necesită ca utilizatorul să perceapă baza de date ca fiind formată din tabele
- ❑ Această percepție se aplică numai la *structura logică* (*Nivelul Extern* și *Nivelul Conceptual* din arhitectura pe 3 nivele a bazei de date), nu și *structurii fizice*, care poate fi implementată utilizând o varietate de modalități de stocare
- ❑ Majoritatea sistemelor moderne sunt bazate pe modelul relațional

# Modelul de date relațional

## ■ Exemplu

Presupunem că din baza de date a unei organizații cu mai multe filiale alegem să reprezentăm datele despre filiale și personalul angajat prin două tabele: *Filiale* și *Angajați*

### *Filiale*

<b><i>NrFil</i></b>	<b><i>Adresa</i></b>	<b><i>Orasul</i></b>	<b><i>CodPostal</i></b>	<b><i>Telefon</i></b>	<b><i>Fax</i></b>
<b>F3</b>	<b>Rozelor 25</b>	<b>Timișoara</b>	<b>1700</b>	<b>121212</b>	<b>121212</b>
<b>F4</b>	<b>Stejeriș 19</b>	<b>Brașov</b>	<b>2200</b>	<b>232323</b>	<b>232323</b>
<b>F5</b>	<b>Eroilor 35</b>	<b>Timișoara</b>	<b>1700</b>	<b>434343</b>	<b>434343</b>
<b>F6</b>	<b>Unirii 10</b>	<b>Focșani</b>	<b>1500</b>	<b>454545</b>	<b>454545</b>



# Modelul de date relațional

## ■ Exemplu

Presupunem că din baza de date a unei organizații cu mai multe filiale alegem să reprezentăm datele despre filiale și personalul angajat prin două tabele

*Angajați*

<b>NrMarca</b>	<b>Nume</b>	<b>Prenume</b>	<b>Adresa</b>	<b>Orasul</b>	<b>Functia</b>	<b>Salariul</b>	<b>NrFil</b>
214	Burcea	Ion	Lalelelor 12	Timișoara	manager	5000	F3
215	Gheorghe	Alina	Cetății 21	Timișoara	contabil	4000	F3
216	Turcea	Elena	Warte 8	Brașov	secretară	3000	F4
217	Vasile	Valentin	Gării 32	Timișoara	portar	200	F5

# Modelul de date în rețea

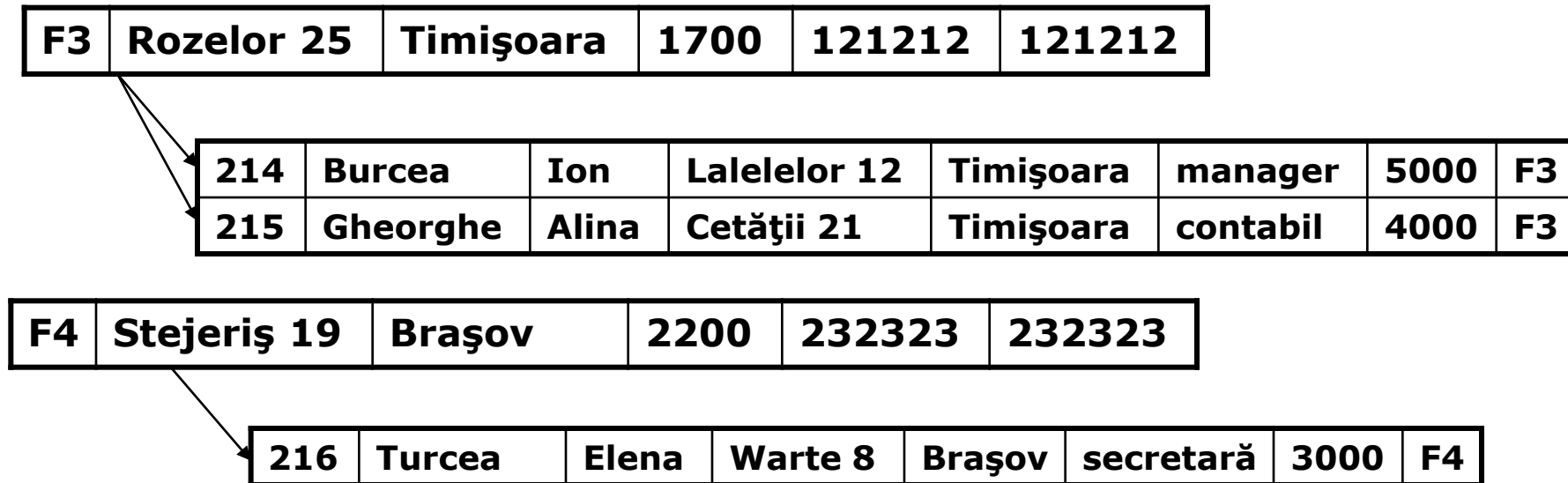
---

- ❑ Datele sunt reprezentate printr-o **colecție de înregistrări**
- ❑ Relațiile sunt reprezentate prin **direcții**
- ❑ Spre deosebire de modelul relațional, aici relațiile sunt modelate explicit prin direcții, care devin pointeri în implementarea propriu-zisă
- ❑ Modelul poate fi asemănat cu o structură de **grafuri**
  - Înregistrările sunt reprezentate ca **noduri**
  - Direcțiile sunt reprezentate ca **muchii**

# Modelul de date în rețea

## ■ Exemplu

Transpunem exemplul anterior într-un model de date în rețea

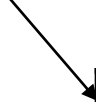


# Modelul de date în rețea

## ■ Exemplu

Transpunem exemplul anterior într-un model de date în rețea

<b>F5</b>	<b>Eroilor 35</b>	<b>Timișoara</b>	<b>1700</b>	<b>434343</b>	<b>434343</b>
-----------	-------------------	------------------	-------------	---------------	---------------



<b>217</b>	<b>Vasile</b>	<b>Valentin</b>	<b>Gării 32</b>	<b>Timișoara</b>	<b>portar</b>	<b>200</b>	<b>F5</b>
------------	---------------	-----------------	-----------------	------------------	---------------	------------	-----------

<b>F6</b>	<b>Unirii 10</b>	<b>Focșani</b>	<b>1500</b>	<b>454545</b>	<b>454545</b>
-----------	------------------	----------------	-------------	---------------	---------------

# Modelul de date ierarhic

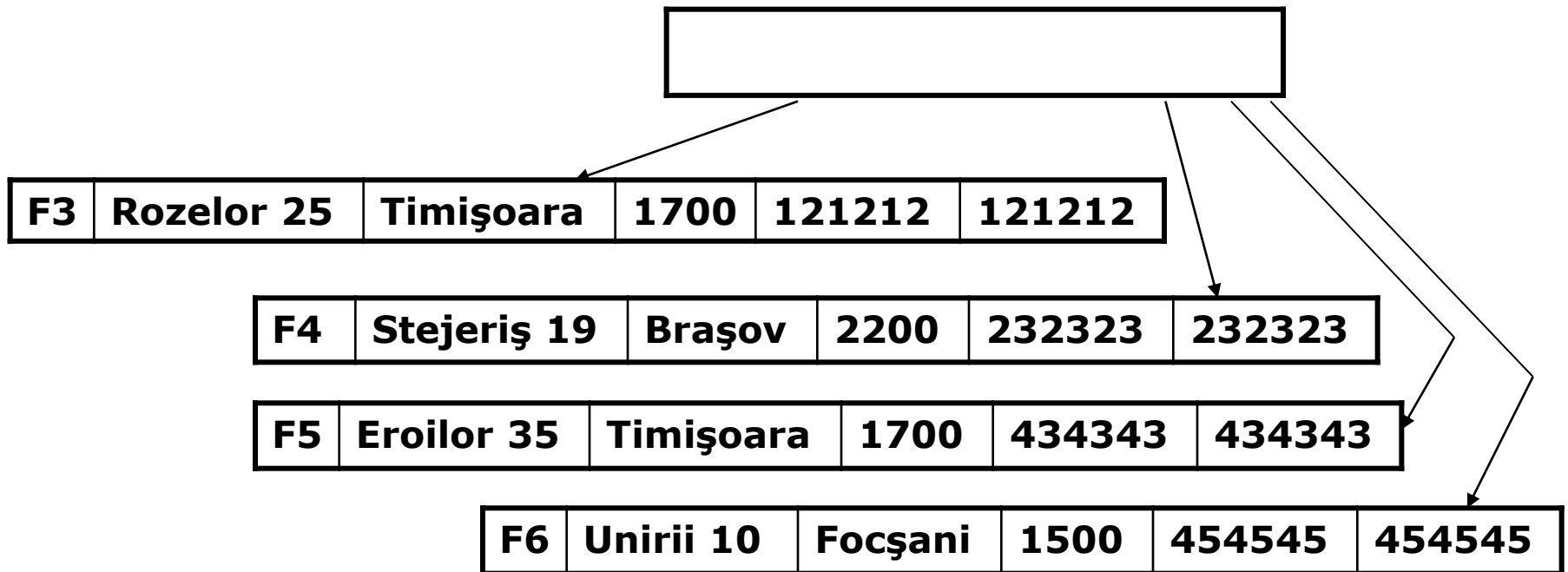
---

- ❑ Constituie un tip restrâns de model în rețea
- ❑ Datele sunt reprezentate printr-o **colecție de înregistrări**
- ❑ Relațiile sunt reprezentate prin **direcții**
- ❑ Permite ca un **nod să posede numai un singur părinte**
- ❑ Poate fi reprezentat ca un graf de tip arbore

# Modelul de date ierarhic

## ■ Exemplu

Transpunem exemplul anterior într-un model de date ierarhic



# Modele de date bazate pe înregistrări

---

- Sunt utilizate pentru a specifica
  - structura generală a bazei de date
  - o descriere de nivel superior a implementării acesteia
- Principalul lor dezavantaj este că nu pun la dispoziție facilități de specificare explicită a constrângerilor asupra datelor
- **Modelele relaționale** adoptă o abordare *declarativă* pentru procesarea bazei de date
  - specifică *ce* date vor fi regăsite
- **Modelele în rețea și ierarhice** adoptă o abordare *navigațională*
  - arată *cum* vor fi regăsite datele

# Modelarea conceptuală

---

- Printr-o examinare a arhitecturii cu trei nivele, se poate observa că schema conceptuală este nucleul bazei de date
- Ea suportă toate vederile externe și este suportată de schema internă
- Schema internă este doar o implementare fizică a schemei conceptuale



# Modelarea conceptuală

---

- ❑ Schema conceptuală trebuie să fie o **implementare completă și corectă a cerințelor companiei** (organizației) privind datele
- ❑ Dacă acest lucru nu este realizat, o serie de informații despre companie vor lipsi sau vor fi reprezentate incorect, ceea ce va crea dificultăți în implementarea completă a uneia sau mai multor vederi externe

# Modelarea conceptuală

---

- Modelarea conceptuală sau proiectarea conceptuală a bazei de date este procesul de construire a unui model de informații utilizate într-o companie
- Este independent de
  - detaliile de implementare, cum ar fi sistemul SGBD
  - programele aplicație, limbajele de programare
  - orice alte tipuri de considerații fizice
- Acest model de date se numește **model de date conceptual**