Curs 10 – Proiectarea sistemelor informatice

Cuprins

- ✓ Proiectarea bazelor de date
- ✓ Proiectarea fișierelor
- ✓ Proiectarea depozitelor de date

Proiectarea bazei de date

- Proiectarea unei baze de date cuprinde proiectarea structurii:
  a) conceptuale
  b) logice
  c) fizice.
- Pot apărea situații când sunt necesare reveniri la activitățile nivelurilor anterioare.
- De exemplu, odată cu proiectarea structurii fizice pot apărea cerințe de modificări în structura conceptuală.
- Suport: **Diagrama claselor**, **Diagrama entitate asociere** , **Diagrama obiectelor**

Alegerea sistemului de gestiune a bazei de date

a) Stabilirea **cerinţelor** beneficiarului de sistem şi studiul acestora sub aspectul:
- tipurilor de structuri de date;
- timpului de răspuns pentru cerinţele respective;
- metodelor de acces;
- confidenţialitate;
- tipul aplicaţiilor;
- obiectivele sistemului etc.

b) stabilirea **criteriilor de alegere a unui SGBD** din cadrul celor candidate, în funcţie de cerinţele beneficiarului.

c) **inventarierea SGBD-urilor existente** şi stabilirea corespondenţei între cerinţele beneficiarului şi caracteristicile SGBD-urilor, astfel încât să fie capabile să satisfacă cel puţin cerinţele prestabilite.

d) **alegerea propriu-zisă a unui SGBD** din cadrul celor candidate, în funcţie de criteriile prestabilite.

Criteriile pentru alegerea unui anumit tip de SGBD

- **Portabilitatea SGBD-ului** – posibilitatea de a utiliza un SGBD de pe un sistem de calcul pe un altul. Portabilitatea cuprinde două aspecte: portabilitatea programelor propriu-zise şi portabilitatea datelor.
- **Costul sistemului**.
- **Facilităţile de implementare**, întreţinere şi exploatare a bazei de date.
- Posibilitatea gestionării **structurilor complexe de date**, cum ar fi cele de tip arborescent sau reţea.
- **Multitudinea metodelor de acces**. În funcţie de cerinţele proprii aplicaţiei, sistemul va trebui să suporte interogări sau actualizări în timp real având proceduri de tip conversaţional.
- **Protecţia şi securitatea datelor** din bază.
- **Specificul aplicaţiei**. Este cunoscut faptul că programele sunt orientate pe aplicaţii, cum ar fi: programarea producţiei, aprovizionare-desfacere, optimizări, prognoze etc.
- **Timpul** necesar **pentru formarea** cadrelor care să utilizeze SGBD-ul.

a) Proiectarea schemei conceptuale

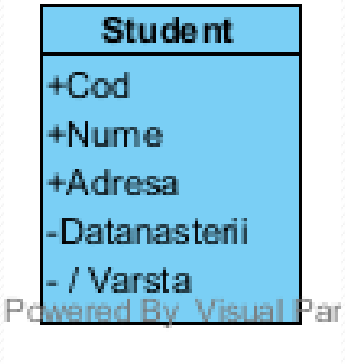
- Este realizată de către **echipa de proiectare a SGBD**. Punctul de plecare în proiectarea structurii conceptuale îl reprezintă colecţiile de date, stabilite în **modelul logic proiectat**, când s-a realizat totodată şi schiţarea unui prim model conceptual de ansamblu al datelor. Proiectarea schemei conceptuale cuprinde activităţile:
  - definirea detaliată a colecţiilor de date;
  - revizuierea legăturilor dintre colecţii;
  - rafinarea modelului conceptual al datelor;
  - transpunerea modelului conceptual al datelor.
- Nu în toate cazurile se impune parcurgerea tuturor **etapelor de normalizare**, însă în mod obligatoriu prima formă normală este obligatorie.
- Necesitatea normalizării progresive este dată de faptul că anumite tabele pot genera o serie de situaţii nedorite, aşa-numitele „**anomalii de actualizare**“.
  - *Anomalia de ştergere* rezultă din faptul că ştergând un tuplu al unei tabele, odată cu ştergerea anumitor informaţii se pierd şi informaţiile utile, existente în tuplul respectiv.
  - *Anomalia de adăugare* rezultă din faptul că nu pot fi incluse noi informaţii într-o tabelă deoarece nu se cunosc şi alte informaţii cerute pentru adăugarea unui nou tuplu la acea tabelă, în principal valorile pentru atributele din cheie.
  - *Anomalia de modificare* rezultă din faptul că este dificil de modificat o valoare a unui atribut atunci când ea apare în mai mult decât într-un tuplu al tabelei.

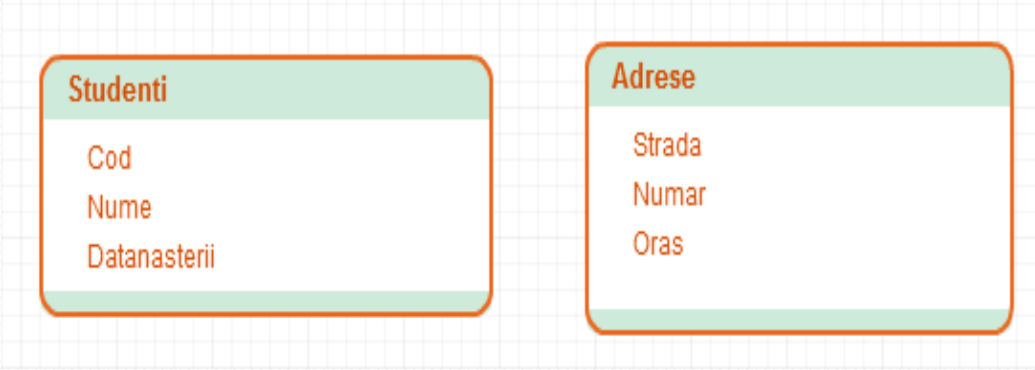
De la modelul conceptual la baza de date relationala

- Transformare 1:1 a claselor in tabele:
  - **Prea multe tabele** – pot rezulta mai multe tabele decat e necesar
  - **Prea multe join-uri** – ca o consecinta a supranormalizarii
  - **Tabele lipsa** – asocierile m:n intre clase implica utilizarea unei tabele special de legatura
  - **Tratarea necorespunzatoare mostenirii**
  - **Denormalizarea tabelelor** – anumite date se regasesc in prea multe tabele

Transformarea claselor in tabele

- **Numele** tabelei reprezinta pluralul numelui clasei
- Toate **atributele simple** sunt transformate in campuri
- **Atributele compuse** devin tabele de sine statatoare
- **Atributele derivate** nu vor avea nici un corespondent in tabela

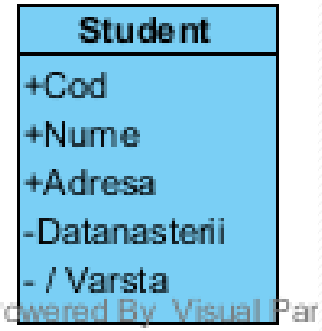


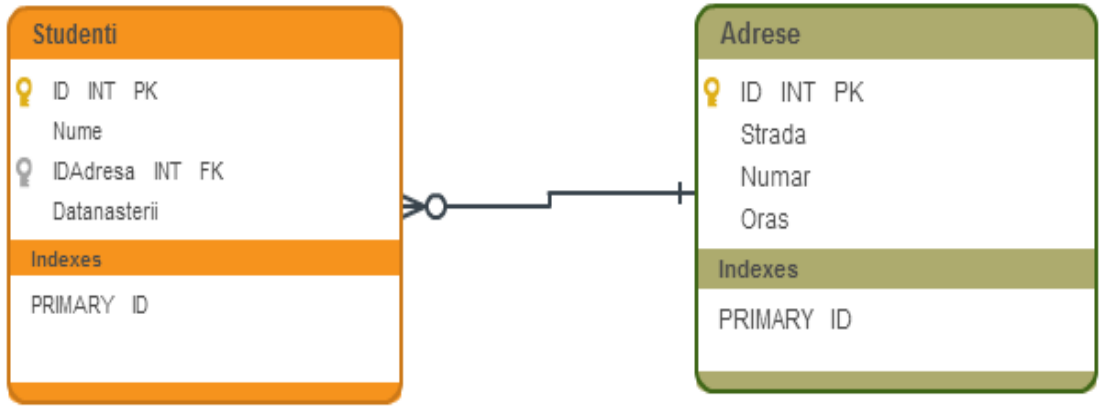


Transformarea claselor in tabele (cont)

- **Chei surogat** – care nu sunt obtinute din domeniul problemei modelate
- Cheia nu poate fi preluata din clasa UML
- **O buna practica**: utilizarea, cand e posibil, a cheilor de tip intreg generate automat de SGBD:
  - Usor de intretinut
  - Eficient (interogari rapide)
  - Simplifica definirea cheilor externe
- **Disciplina de proiectare a BD**:
  - Toate cheile surogat vor fi numite **ID**
  - Toate cheile externe se vor numi **<NumeTabel>ID**

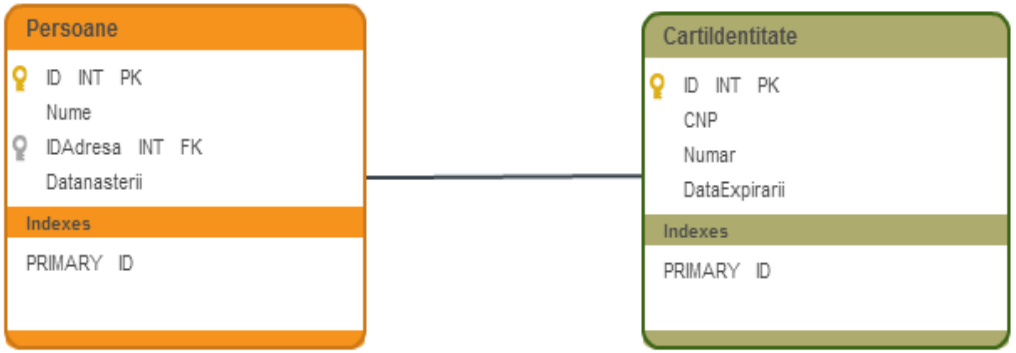
Transformarea claselor in tabele (cont)





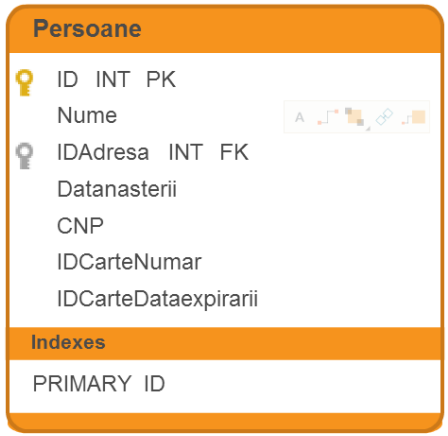
Transformarea asocierilor simple

- **1:0,1**
  - Se creeaza cate o tabela corespunzatoare fiecarei clase implicate in asociere
  - Cheia tabelei corespunzatoare multiplicitatii “1,1”este cheie externa in a doua tabela
  - O singura cheie va fi generata automat, de obicei cea corespunzatoare multiplicitatii 1



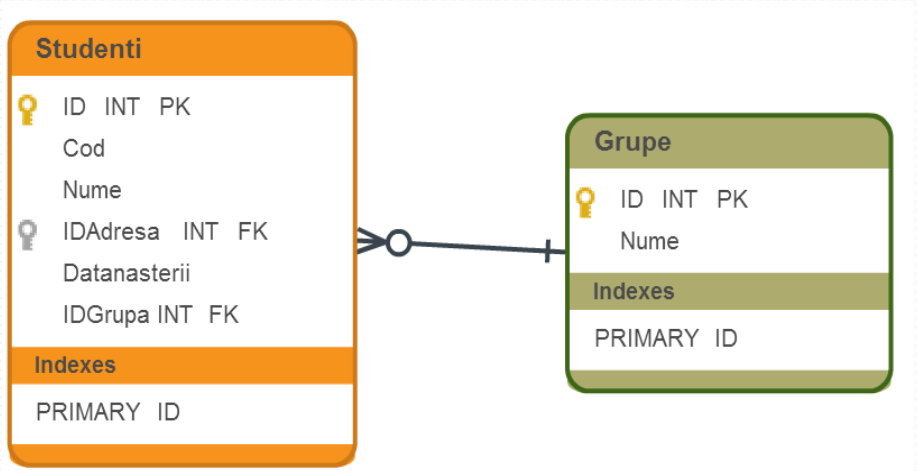
Trasformarea asocierilor simple (cont)

- **1:1**
  - Se creeaza o singura tabela ce contine atributele ambelor clase associate
  - Aceasta varianta de transformare se aplica si asocierilor “1:0,1”atunci cand este vorba de un numar relativ mic de cazuri in care obiectele primei clase nu sunt legate de obiectele celei de-a doua



Trasformarea asocierilor simple (cont)

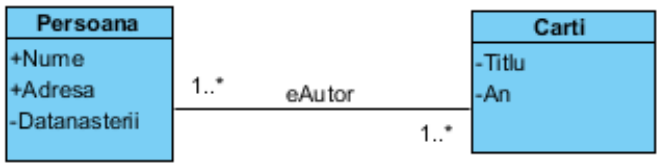
- 1:1..*
  - Se creeaza cate o tabela corespunzatoare fiecarei clase implicate in asociere
  - Cheia tabelei corespunzatoare multiplicitatii “1”este cheia externa in a doua tabela, corespunzatoare multiplicitatii “1..*”

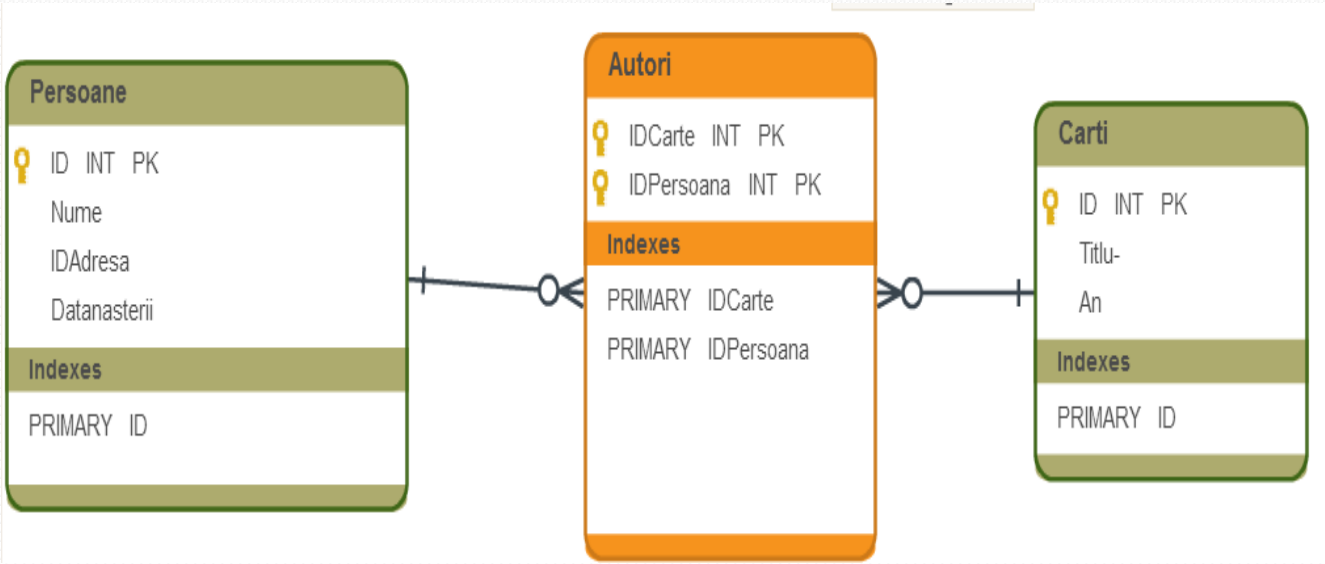


Trasformarea asocierilor simple (cont)

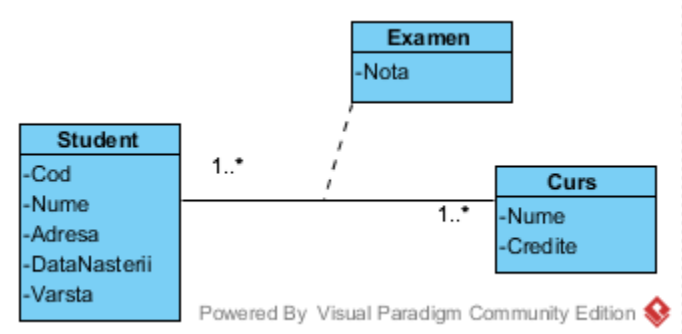
- **1..*:1..***
  - Se creeaza cate o tabela corespunzatoare fiecarei clase implicate in asociere
  - Se creeaza **o tabela aditionala de legatura**
  - Cheile primare corespunzatoare tabelelor initiale sunt definite ca si chei externe in tabela de legatura
  - Cheia primara a tabelei de legatura este de obicei compusa din cele doua chei externe spre cele doua tabele. Sunt cazuri in care se utilizeaza si aici cheie surogat
  - Daca asocierea contine o **clasa asociere**, toate atributele acestei clase vor fi inserate in tabela de legatura
  - De obicei, numele tabelei de legatura este o combinatie a numelor tabelelor initiale

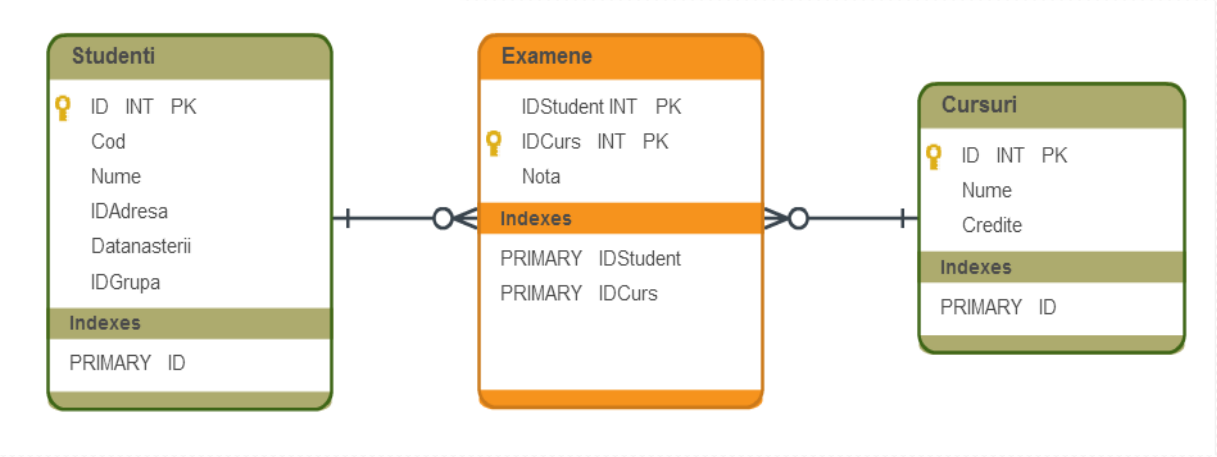
Trasformarea asocierilor simple (cont)





Trasformarea asocierilor simple (cont)

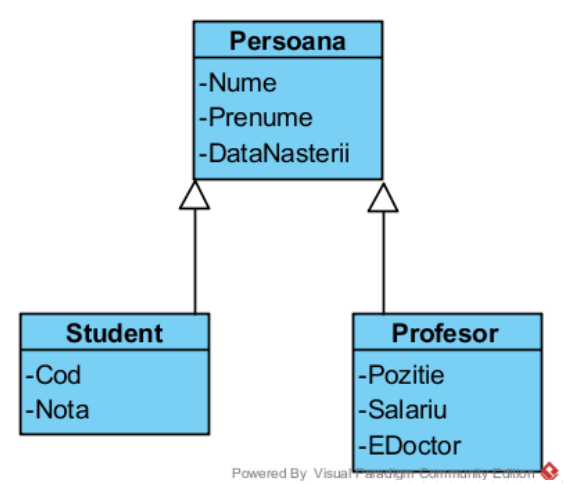


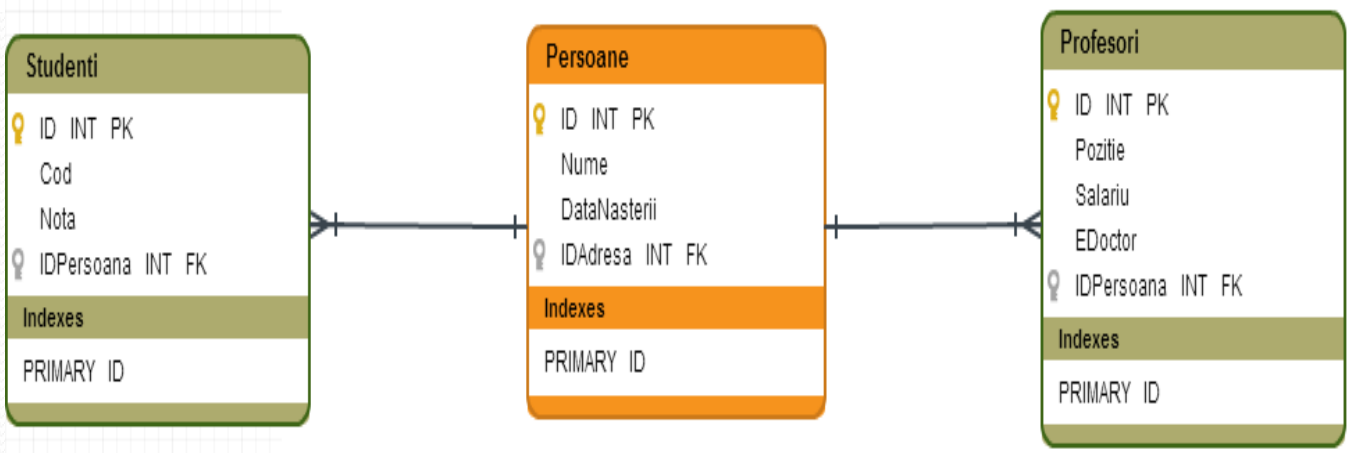


Transformarea mostenirii

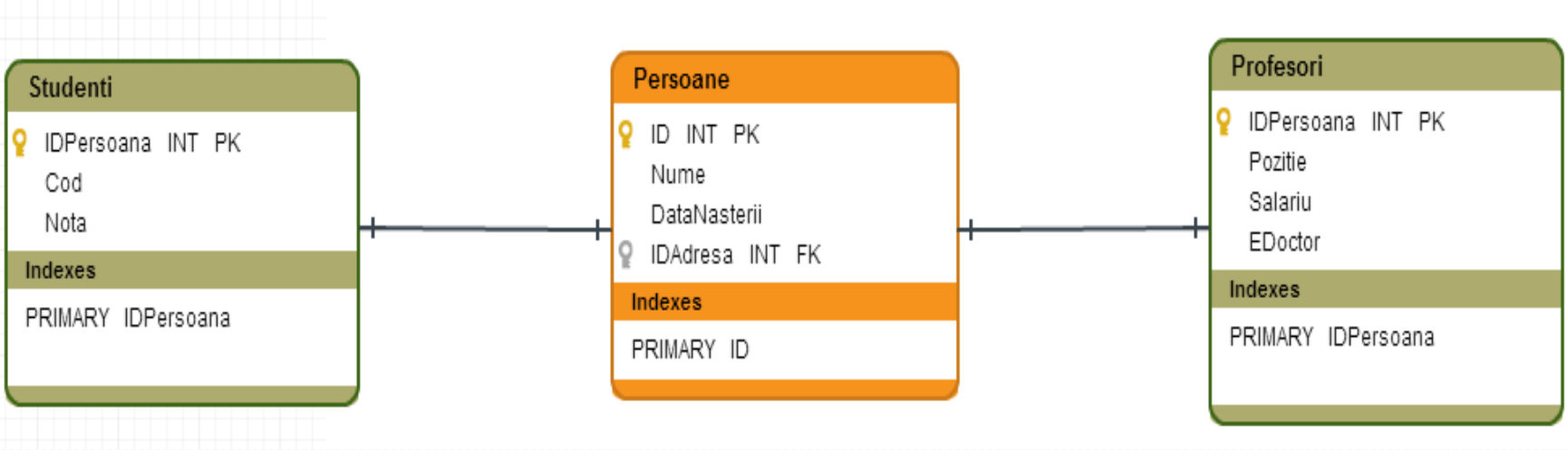
- **Alternativa 1**
  - Presupune crearea cate **unei tabele pentru fiecare clasa parinte si copil** si **cate un view pentru fiecare pereche parinte-copil**
  - Flexibilitate – permite adaugarea de noi subclase fara impact asupra claselor sau view-lor existente
  - Implica crearea celor mai multe tabele/ view-uri
  - Posibile probleme de performanta deoarece fiecare consultare presupune executia unui joim

Transformarea mostenirii (cont)





Transformarea mostenirii (cont)

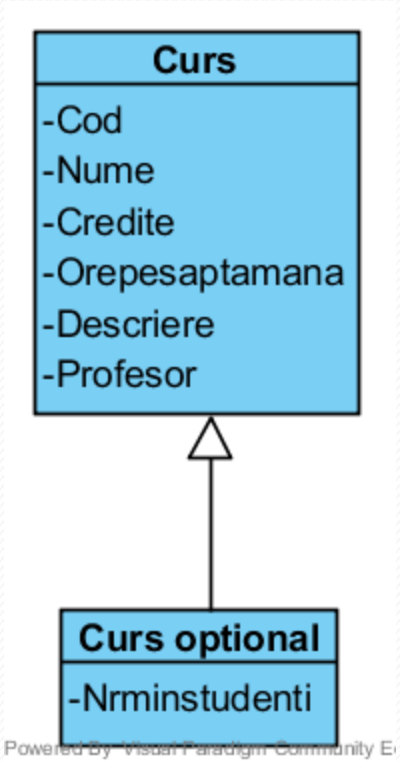


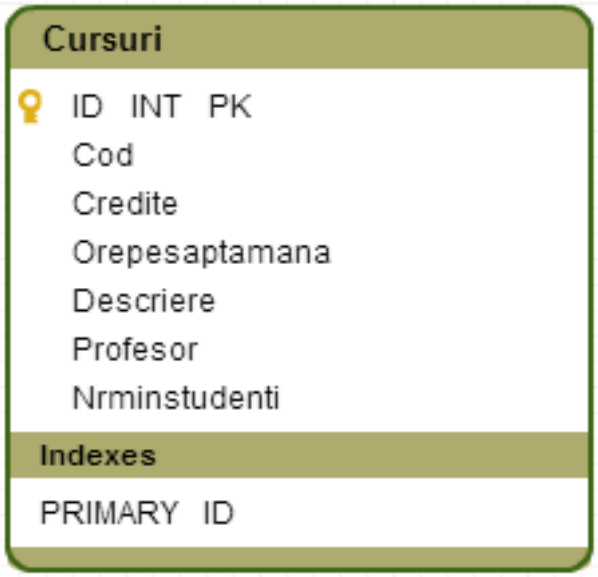
CREATE VIEW StudentiComplet... AS SELECT Persoane.*, Cod, Nota FROM Persoane INNER JOIN Studenti ON Persoane.ID=Studenti.IDPersoana

Transformarea mostenirii (cont)

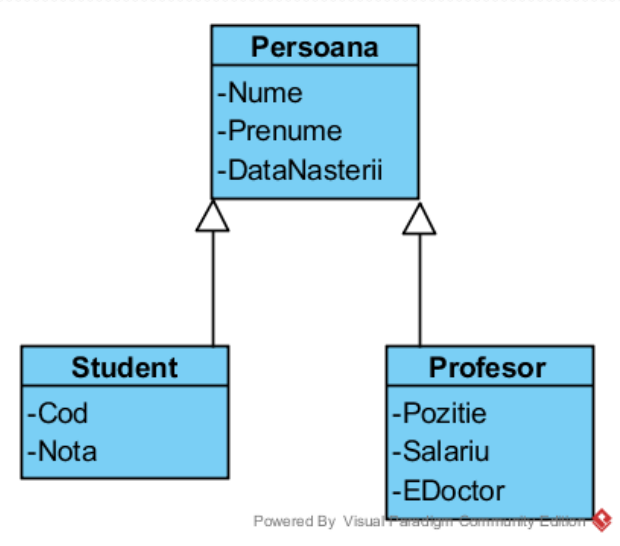
- **Alternativa 2**
  - Se creeaza **o singura tabela** corespunzatoare superclasei si se denormalizeaza toate atributele corespunzatoare subclaselor
  - Implica crearea celor mai putine tabele/view-uri – optional se pot defini o tabela de subclase si view-uri corespunzatoare pentru fiecare subclasa
  - Implica cea mai buna performanta
  - Adaugarea unei subclase noi implica modificari structurale
  - Creste “artificial” spatial utilizat

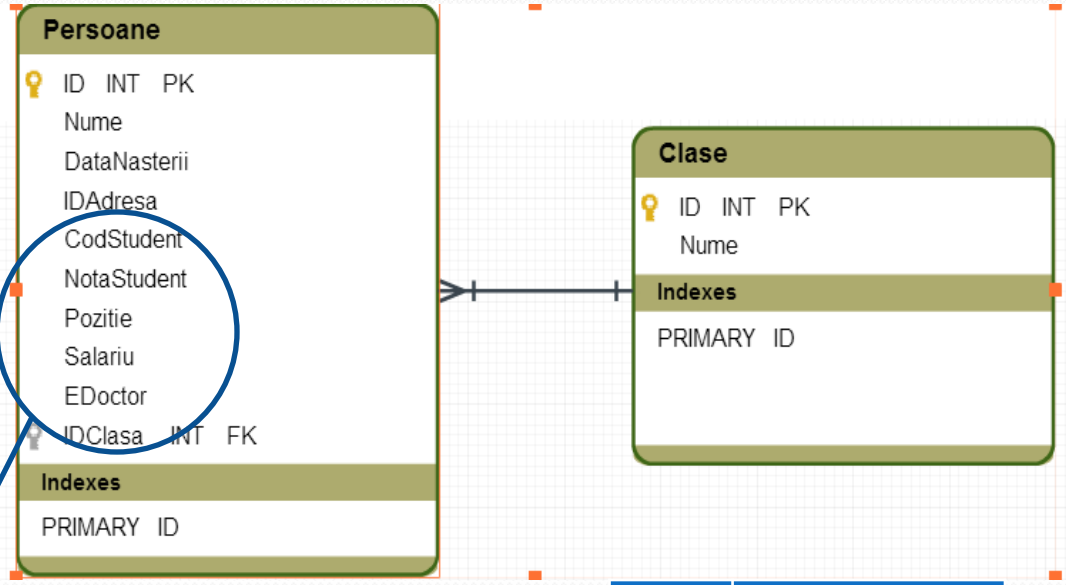
Transformarea mostenirii (cont)





Transformarea mostenirii (cont)





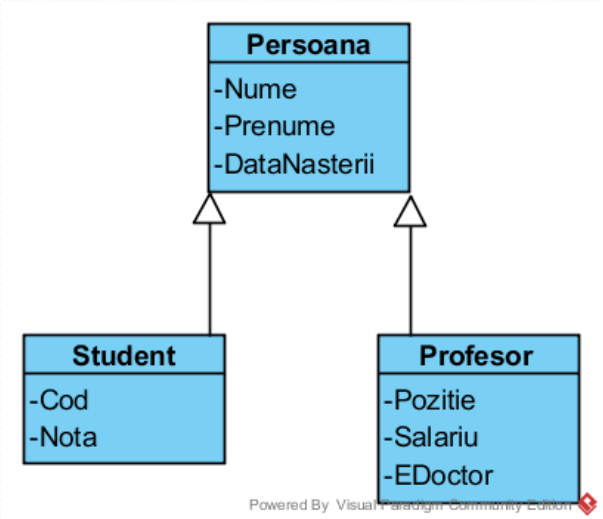
CREATE VIEW Studenti... **AS SELECT** ID, Nume, IDAdresa,DataNasterii, CodStudent, NotaStudent **FROM** Persoane **WHERE** IDClasa=2

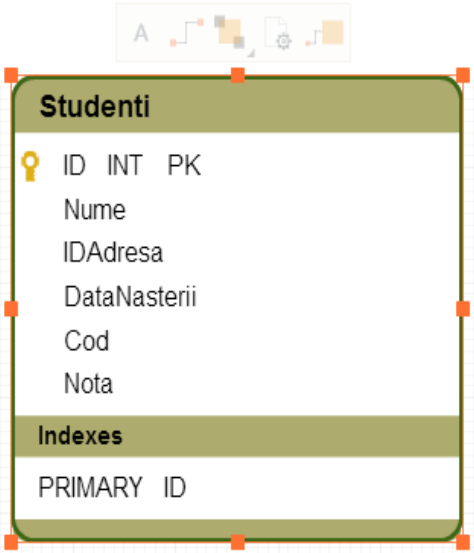
Id	Nume
1	Necunoscut
2	Student
3	Profesor

Transformarea mostenirii (cont)

- **Alternativa 3**
  - Presupune crearea **cate unei tabele pentru fiecare subclasa si denormalizarea atributelor superclasei in fiecare din tabelele create**
  - Performanta obtinuta este satisfacatoare
  - Adaugarea unei noi subclase nu implica modificari structurale
  - Posibilele modificari structurale la nivelul superclasei afecteaza toate tabelele definite

Transformarea mostenirii (cont)





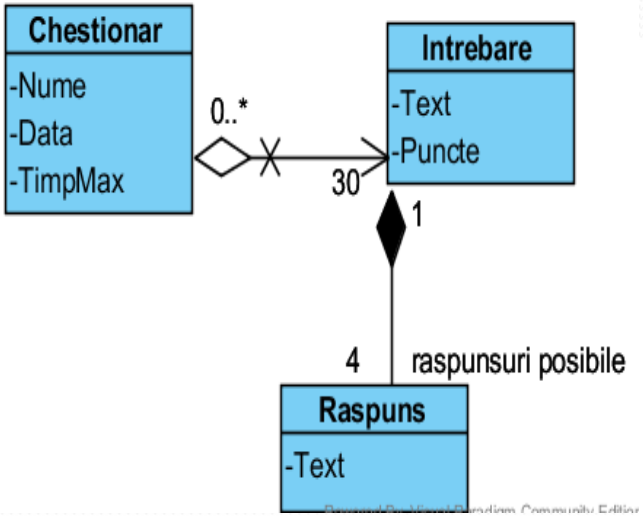
Transformarea mostenirii (cont)

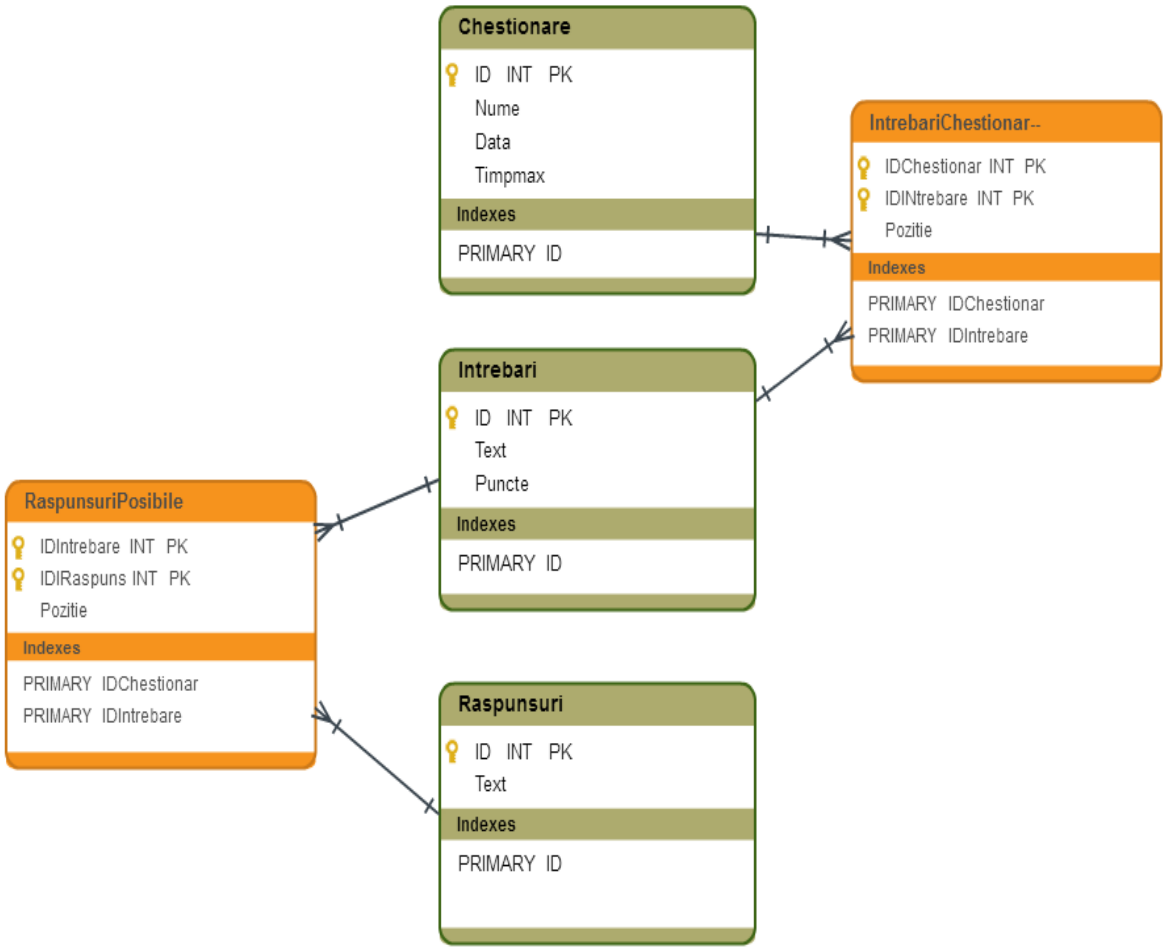
- Care este alternativa potrivita?
- Daca **numarul inregistrarilor stocate in tabele este redus** (nu sunt probleme de performanta) poate fi selectata cea mai flexibila varianta, **Alternativa 1**
- Daca superclasa are un **numar restrans de atribute** (comparativ cu subclasele) atunci alternative potrivita este **Alternativa 3**
- Daca **subclasele au instante putine**, atunci cea mai buna este utilizarea **Alternativei 2**

Transformarea agregarii/ compunerii

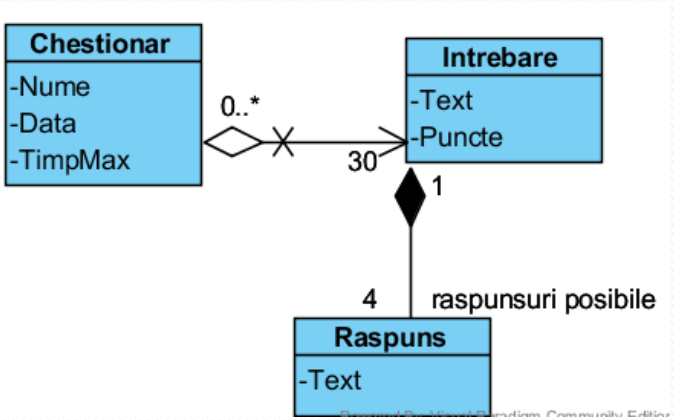
- Agregarea si compunerea sunt modelate in mod similar asocierilor
- In cazul relatiilor de compunere de obicei se utilizeaza o singura tabela de legatura deoarece compunerea implica mai multe relatii 1:1
- **Numarul fix** de *parti* intr-un *intreg* presupune introducerea unui numar egal de chei externe in tabela *intreg*
- In cazul implementarii compunerii in tabele separate este necesara setarea ***stergerii in cascada*** *(in cazul agregarii acest lucru nu este necesar)*

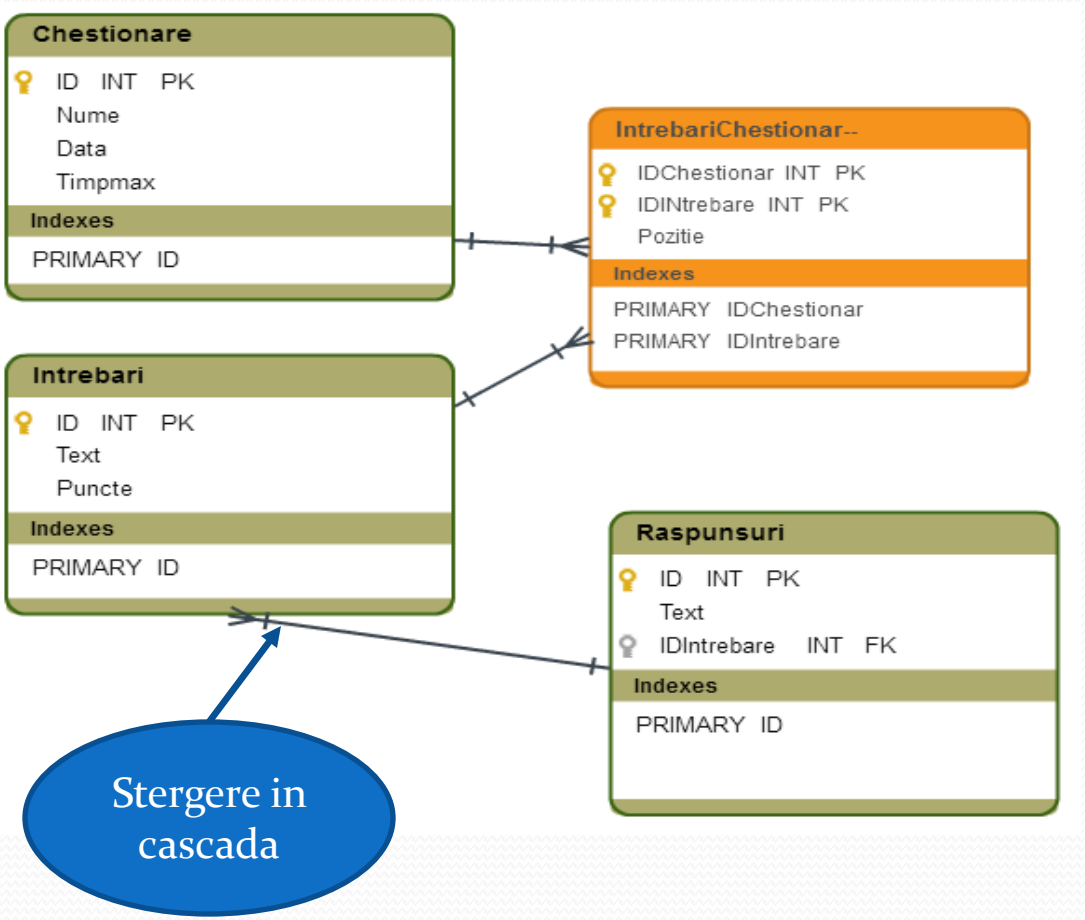
Transformarea agregarii/ compunerii (cont)



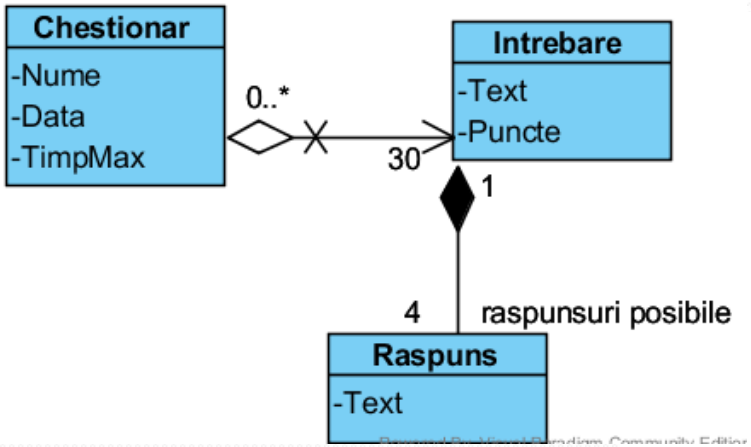


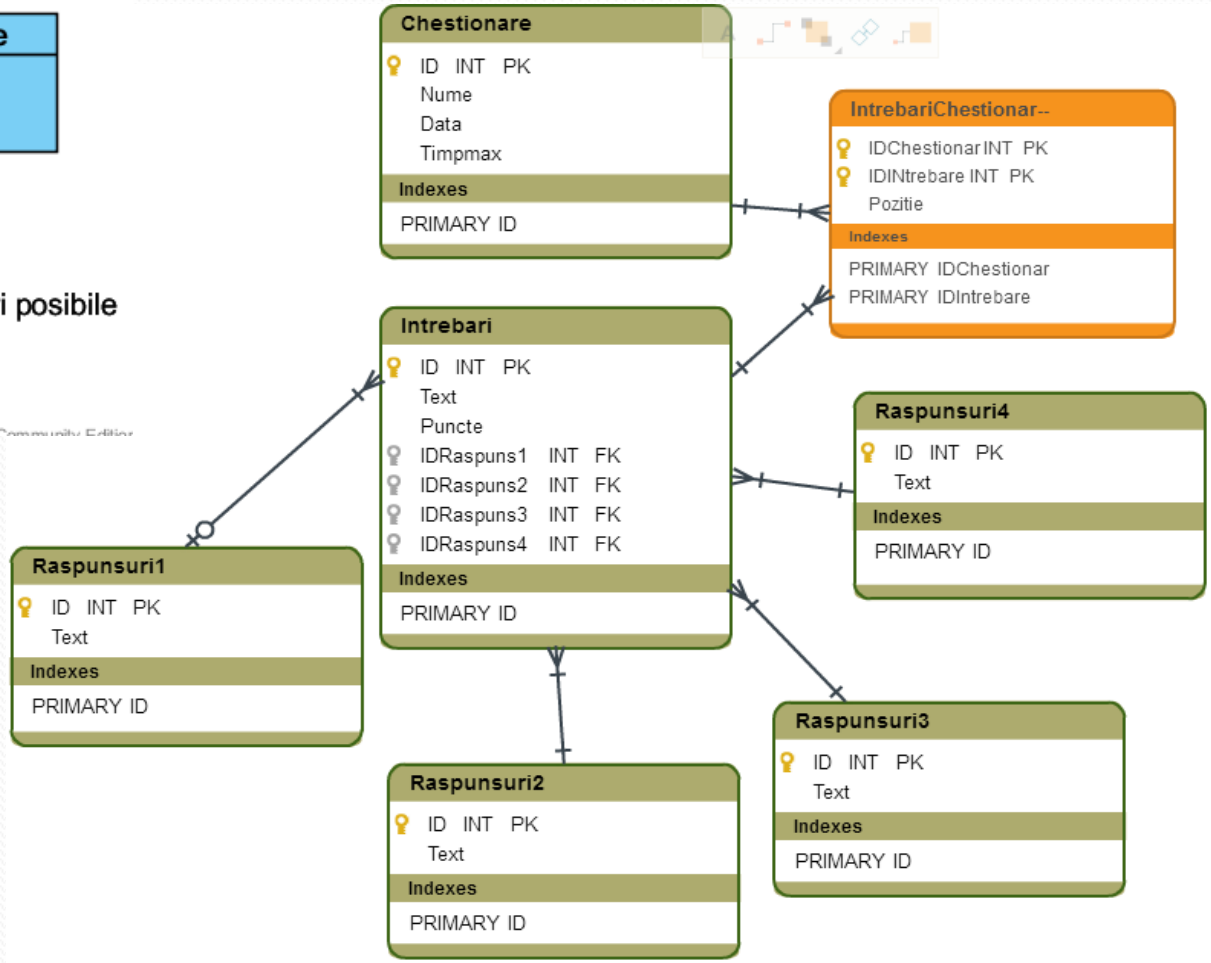
Transformarea agregarii/ compunerii (cont)





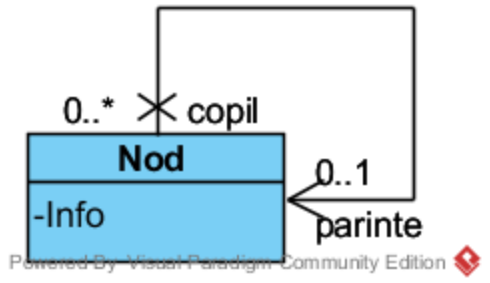
Transformarea agregarii/ compunerii (cont)

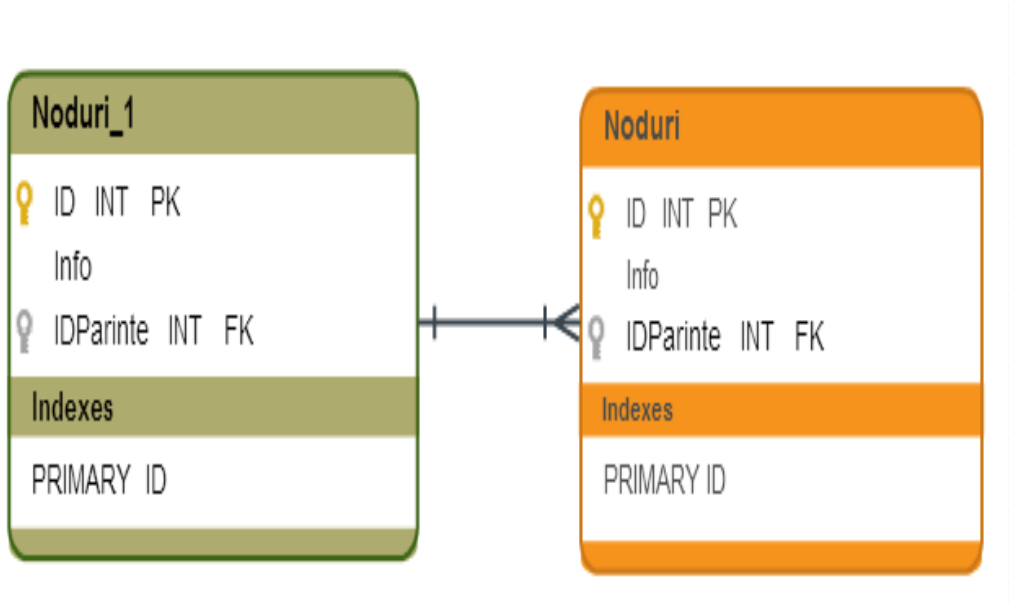




Transformarea autoasocierilor

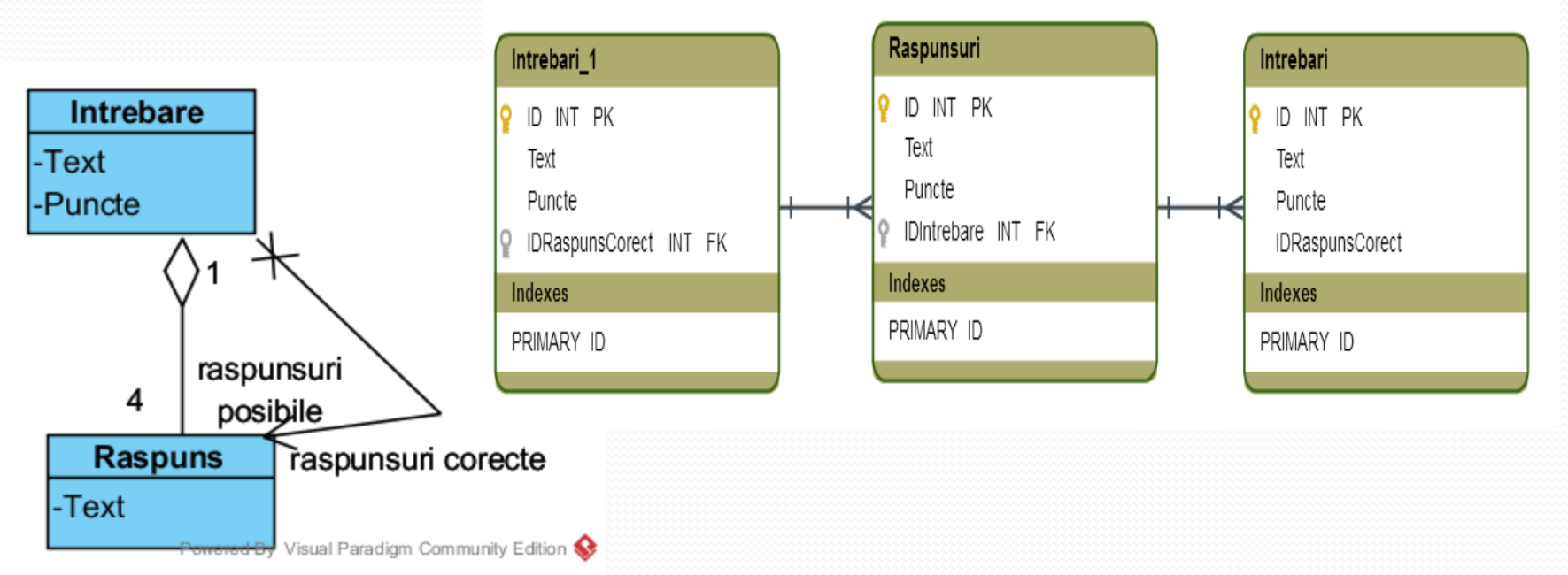
- Se introduce o cheie externa care pointeaza spre aceeasi clasa
- Daca este setata proprietatea **stergerii in cascada** exista 2 inregistrari care se refera reciproc, stergerea uneia din ele va genera o **eroare**





Transformarea autoasocierilor (cont)

- **Stergerea in cascada** genereaza o problema similara si in cazul a doua tabele care se refera reciproc



Proiectarea schemei logice/externe

- **Structura logica a bazei** de date reprezintă forma sub care apare structura conceptuală a bazei de date **pentru un utilizator oarecare**. Programele de aplicaţie operează asupra elementelor structurii conceptuale prin intermediul structurii logice, având acces doar la acele elemente ale structurii conceptuale care sunt incluse în structura logică.
- În general elementele care compun structura logică sunt similare celor care compun structura virtuală, totuşi **depind de tipul de SGBD utilizat**.
- Deci am putea spune că prin **separarea nivelului logic de nivelul conceptual se separă funcţia de administrare de funcţia de utilizare** – exploatare a bazei de date.
- Practic este vorba de:
  - **identificarea viziunilor** (view-urilor) care vor fi construite. Viziunea reprezintă o tabelă stocată logic (nu şi fizic) care preia anumite câmpuri din una sau mai multe tabele care îndeplinesc anumite condiţii. Prin utilizarea viziunilor, utilizatorii primesc acces doar asupra unei părţi din baza de date
  - **identificarea drepturilor de acces** asupra tabelelor şi viziunilor pentru fiecare tip de utilizator (tipuri de drepturi: select, insert, update, delete)

Proiectarea schemei fizice a BD

- Spre deosebire de structura conceptuală, care îmbracă diferite forme de reprezentare (liniară, arborescentă, reţea, relaţională) memorarea datelor **pe suportul fizic** îmbracă numai forma unei **structuri liniare**. De aceea se pune problema liniarizării structurii virtuale.
- **Metoda de liniarizare a structurii** virtuale este **specifică diferitelor SGBD** utilizate. Astfel, o serie de SGBD-uri utilizează pentru realizarea structurii fizice **metode tipice** de organizare a informatiilor folosite **în cadrul sistemelor de operare gazdă**.
- Alte SGBD-uri utilizează pentru realizarea structurii fizice **metode proprii** de organizare a informaţiilor, independente de metodele folosite în cadrul sistemului de operare gazdă.
- Cea de-a doua grupă de SGBD-uri depinde mai puţin de sistemele de operare gazda, ceea ce le imprimă o portabilitate sporită comparativ cu cele din prima categorie.
- Practic, **se estimează spaţiu ocupat de fiecare tabelă** (dependent de SGBD-ul ales) ca sumă între:
  - numărul fix de octeţi alocat pentru memorarea structurii tabelei (aproximăm pentru fiecare tabelă 20 bytes) 20 * 13 = 260 bytes
  - numărul variabil de octeţi estimat ca produs între:numărul estimat de înregistrări*spaţiul ocupat de o înregistrare

Proiectarea bazei de date -Exemplu

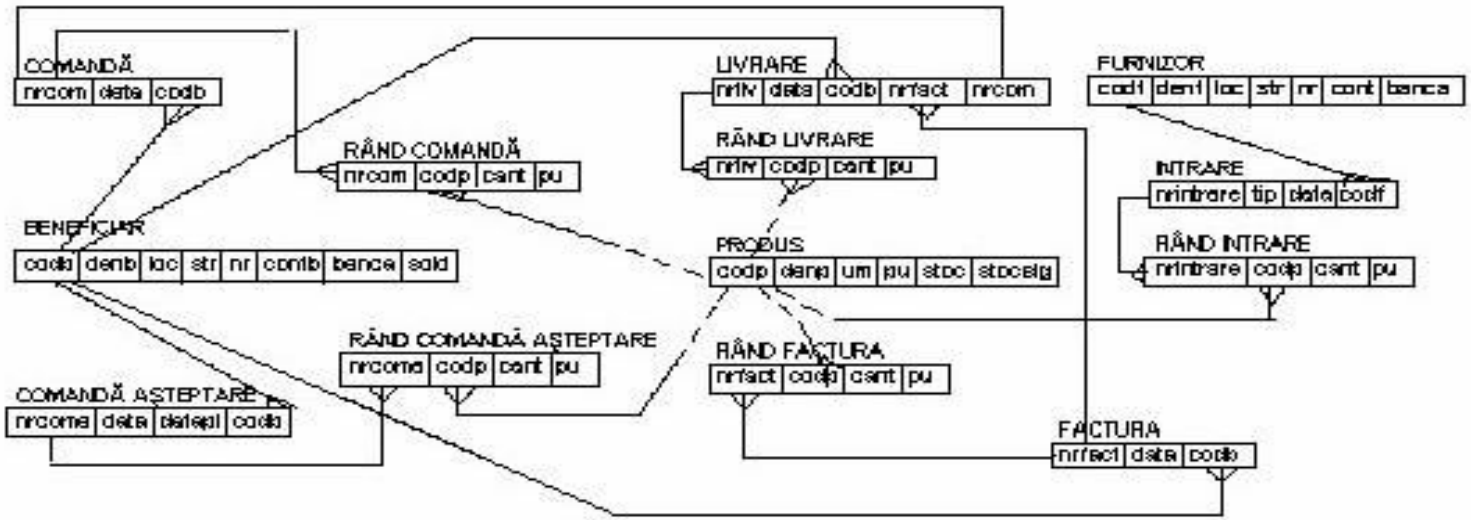
COMANDĂ

Nr. crt.	Denumire informaţie	Nume simbolic	Tip dată	Lungime	Cheie principală	Cheie externă
1	Număr comandă	Nrcom	number	5	X	
2	Data	Data	date	8		
3	Cod beneficiar	Codb	number	5		X
Total				18		

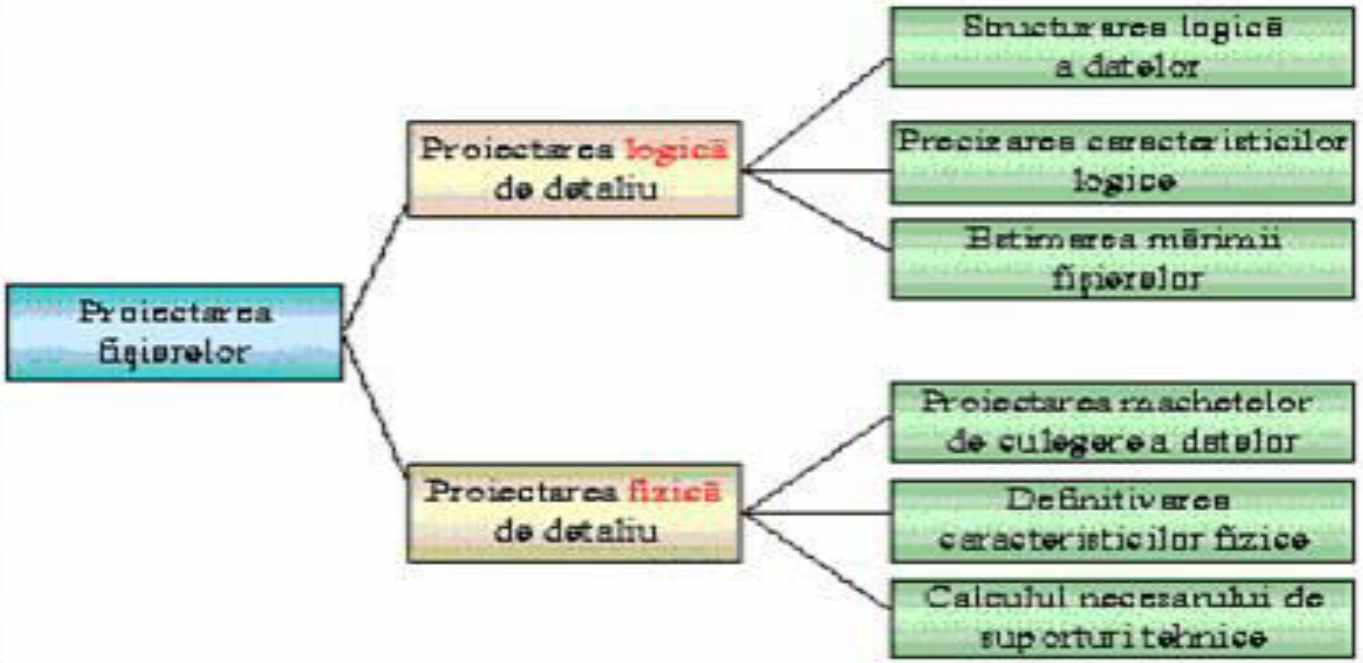
RÂND COMANDĂ

Nr. crt.	Denumire informaţie	Nume simbolic	Tip dată	Lungime	Cheie principală	Cheie externă
1	Număr comandă	Nrcom	number	5		X
2	Cod produs	Codp	number	5		X
3	Cantitate	Cant	number	10		X
4	Preţ unitar	Pu	number	15,5		
Total				35		

Legăturile între tabelele bazei de date:



Proiectarea fisierelor



Proiectarea logică a fişierelor

- Din punct de vedere logic, **fişierul** este o mulţime ***omogenă*** de date identificabilă ca un tot unitar pe suportul fizic.
- Unitatea structurală de bază a fişierului, din punct de vedere logic, este **înregistrarea logică (articolul).** În timp ce sistemul de calcul tratează articolul ca o entitate nediferenţiată, pentru utilizator, el este structurat după un model cu ajutorul **caracteristicilor (câmpurilor).**
- Această fază are ca scop tocmai conturarea articolelor ca unităţi logice de prelucrare a datelor din fişiere. Pentru acest lucru, **proiectarea logică** presupune realizarea a trei subactivităţi:
  I. **Structurarea logică a datelor**
  II. **Stabilirea caracteristicilor logice a datelor**
  III. **Estimarea volumului de date din fişiere**

I. Structurarea logică a datelor

- **Structurarea** datelor în fişiere este operaţia de definire a **structurii logice**, care dă un model de dispunere a datelor în colecţia de date. Structura logică descrie conţinutul informaţional al fişierului pe unitatea informaţională de bază, care este înregistrarea logică (articolul). Această structură este dată printr-o **succesiune de câmpuri cu un anumit format de descriere a unor valori posibile.**
- Proiectarea structurii logice constă în:
  a) Stabilirea **elementelor informaţionale (câmpurile)** care compun înregistrarea logică.
  b) Luarea în considerare a **conţinutului informaţional** real al datelor şi al **cerinţelor informaţionale de prelucrare**.
  c) Precizarea **atributelor logice de utilizare şi reprezentare** a datelor pe suportul tehnic.
  d) Luarea în considerare, din punct de vedere logic, a **posibilităţilor tehnice** oferite de echipamentele periferice.

Câmpurile de date

- **Câmpurile de date** utilizate în structura logică sunt de diferite tipuri, în funcţie de rolul lor în procesul de prelucrare. Cele mai întâlnite tipuri sunt:
  1. ***Date de regăsire*** folosite drept chei de acces.
  2. ***Date de identificare*** care exprimă în clar elemente codificate sau nu.
  3. ***Date de grupare*** care apar sub formă de coduri. Acestea, fiind scurte, faţă de denumirile în clar cărora le sunt ataşate, se folosesc pentru acces, ordonări, regăsiri, legături etc.
  4. ***Data descriptive*** care sunt şiruri de caractere ce descriu în clar caracteristici, proprietăţi, însuşiri ale datelor. Sunt câmpuri de lungime mare având rol explicativ.
  5. ***Date calendaristice*** care exprimă termene şi perioade de timp pentru diferite fapte, evenimente, procese, fenomene etc. Pot fi în diferite formate care conţin ziua, luna şi anul.
  6. ***Date cantitative/valorice*** care cuantifică diferite elemente.
  7. ***Date de legătură*** care au rolul de a înlănţui înregistrările din acelaşi fişier sau din fişiere diferite.
  8. ***Date de stare*** care au rolul de a preciza, prin diferite valori, care este starea înregistrării la momentul execuţiei curente (active, anulate, şterse logic etc.)

Atributele logice de utilizare şi reprezentare a datelor

- sunt elemente referitoare la fişier, care se va afla pe un anumit periferic, descrise în program.
- Fiecare **limbaj** de programare are **instrucţiuni specifice** pentru a descrie aceste atribute logice, în funcţie de fişier şi de perifericul pe care se află.
- Aceste atribute care descriu fişierul sunt:
  a) tipul de înregistrare,
  b) lungimea înregistrării,
  c) etichetele înregistrării,
  d) numele logic de referire a înregistrării.
- **Nu întotdeauna** şi nu în toate limbajele de programare aceste atribute trebuie **descrise în program explicit**.

Posibilităţile tehnice ale echipamentelor periferice

- Aceste caracteristici ale echipamentelor periferice sunt descrise în program **pentru fiecare fişier** existent. Orice operaţie de intrare (citire) sau ieşire (scriere) referitoare la un fişier implică posibilităţile tehnice oferite de perifericul pe care se află fişierul.
- Ele se pot referi atât la **sistemul de calcul** din care fac parte perifericele, cât şi la **perifericele propriu-zise**.
- Descrierea, din punct de vedere logic a acestor posibilităţi, se referă la:
  a) numele logic de referire al fişierului,
  b) mod de organizare şi acces al fişierului,
  c) chei de acces la datele din fişier,
  d) zone tampon de memorie internă (buffere) aferente fişierului.
- În anumite limbaje de programare o parte dintre aceste caracteristici este **implicită**, nefiind necesar să fie descrisă în program.

II. Caracteristici logice - la nivel de înregistrare

1. ***Natura datelor***
2. ***Tipul datelor***
3. ***Mărimea datelor***
4. ***Factorul de repetativitate***
5. ***Gruparea datelor***
6. ***Modul de control***

Atributele de mai sus exprimă forma de utilizare a datelor din fişiere şi ajută la stabilirea următoarelor **aspecte**:

- *Determinarea mărimii în caractere (**lungimii**) unei înregistrări*
- *Stabilirea **formatului** de înregistrare*
- *Stabilirea, parţial sau total, a **condiţiilor de validare logică*** a datelor şi strâns legat de aceasta a erorilor generate şi modul de corectare a lor.
- *Definirea unor **condiţii care apar în prelucrarea** datelor*: manipularea fişierelor (ordonare, interclasare, selecţie etc.), algoritmi de calcul, afişarea rezultatelor.

II. Caracteristicile logice la nivel de fişier

- Cu ajutorul acestora se specifică **înlănţuirea înregistrărilor** ce vor fi prelucrate, sunt următoarele:
  1. ***Forma legăturii între date*** specifică prin atributele sale tipurile de înlănţuiri între înregistrări: liniară, arborescentă, reţea sau fără înlănţuire.
  2. ***Sensul de parcurgere a legăturilor*** poate fi: direct (FIFO - tip coadă), invers (LIFO - tip stivă) sau în ambele sensuri.
  3. ***Modul de exploatare a legăturilor*** poate fi prin descompunere (explozie) sau compunere (implozie).
  4. ***Modul de organizare a datelor în fişier*** poate fi standard (corespunzător fişierelor standard de intrare/ieşire), clasice (secvenţiale, indexat-secvenţiale, relative), speciale (multiindexate, inverse etc.)
  5. ***Modul de acces*** la datele dintr-un fişier poate fi: secvenţial, direct (selectiv), dinamic.
  6. ***Formatul înregistrărilor*** pe care le conţine un fişier poate fi: fix, variabil sau nedefinit.
  7. ***Dinamica datelor din fişier*** se referă la evoluţia în timp a datelor şi este dată prin perioadele de actualizare şi reorganizare a fişierului.

III. Mărimea fişierelor

- **Mărimea fişierelor** se exprimă în bytes (sau multiplii) şi este dată de numărul de caractere ce se estimează că se vor găsi în fişier.
- Pentru aceasta se foloseşte **mărimea în caractere a înregistrării** (**LI**), determinată în urma stabilirii caracteristicilor logice la nivel de înregistrare.
- În continuare se realizează o estimare aproximativă a **numărului maxim posibil de înregistrări** (**NRI**) din fişier. Această estimare se face pe baza experienţei beneficiarului şi a tendinţei de dezvoltare a activităţii sistemului care se proiectează.
- Cu aceste elemente se poate estima mărimea fişierului (MF) astfel: **MF = LI * NRI.**

Indicatorii de activitate ai fişierului

- Dau o imagine asupra aspectelor dinamice pe care le au datele. Iată câţiva indicatori mai des folosiţi:

1. **Indicatori pentru gestiunea datelor** din fişiere. Se estimează la momentul proiectării logice:
   - ***Numărul de înregistrări actualizate*** în fişier pe un interval de timp (n_a - adăugate, n_m - modificate, n_s - şterse). Acest indicator oferă o imagine asupra mărimii fişierului de tranzacţii (mişcări).
   - ***Numărul de înregistrări accesate*** la momentul unei prelucrări automate (n_e).
   - ***Numărul de înregistrări selectate*** pentru a fi utilizate la momentul unei prelucrări automate (n_u, unde $n_u <= n_e$).

- Pe baza acestor indicatori elementari se pot calcula o serie de indicatori derivaţi.

Indicatorii de activitate ai fişierului (cont)

2. Indicatorii **derivaţi** pun în evidenţă anumite caracteristici logice, proprietăţi specifice fiecărui fişier.
   - ***Ponderea înregistrărilor actualizate*** - influenţează alegerea modului de organizare a fişierului şi modul de prelucrare.
   - ***Indicele de mişcare a înregistrărilor*** indică evoluţia într-o perioada de timp timp a mărimii fişierului:

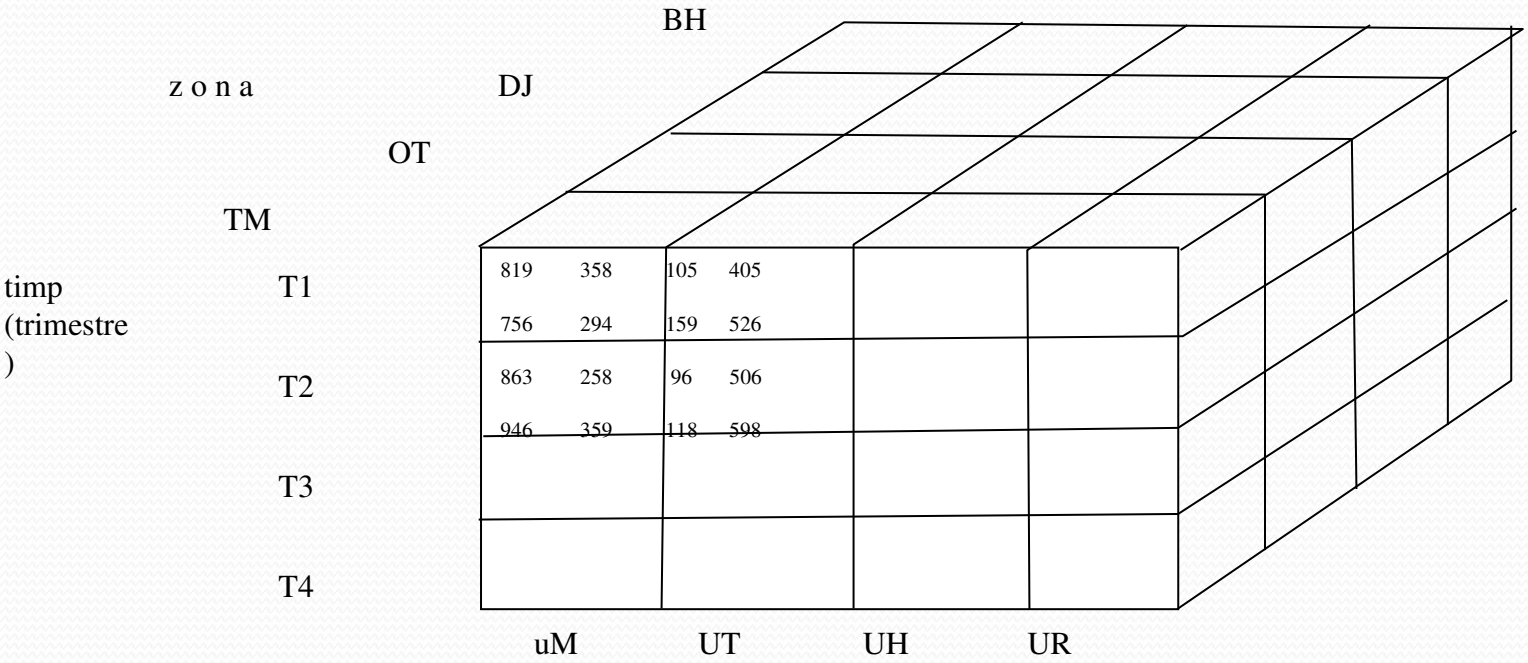
     (Numărul de înregistrări adăugate (na) - Numărul de înregistrări şterse (ns)) / Numărul total de înregistrări din fişier
     - Pentru ca rezultatul să fie în număr de caractere se ia în considerare şi mărimea în caractere a unei înregistrări (LI).
     - Dacă indicele este **pozitiv**, fişierul **se va mării** (expandabilitate). Dacă indicele este **negativ**, este indicat faptul că fişierul **se va micşora** (volatilitate). Dacă indicele este **zero**, fişierul este **stabil**.
   - ***Indicele de utilizare a înregistrărilor*** din fişier arată câte înregistrări sunt selectate în raport cu cele accesate pentru prelucrările efectuate într-o perioadă de timp
   - Acest indicator, împreună cu cel referitor la ponderea actualizărilor, ajută proiectantul de sistem la alegerea modului de acces şi a modului de organizare pentru fişiere.

Depozite de date-modele multidimensionale

- Depozitele de date și instrumentele OLAP sunt bazate pe modele multidimensionale de date. Aceste modele vizualizează datele sub forma unui cub de date (data cub).
- Cubul de date permite modelarea și vizualizarea datelor în dimensiuni multiple. El este definit prin **dimensiuni** și **fapte**.
- În termeni generali, **dimensiunile** exprimă perspectivele în care o anumită organizație dorește să păstreze înregistrarile privitoare la tranzacțiile desfășurate.

DD-cubul 3D

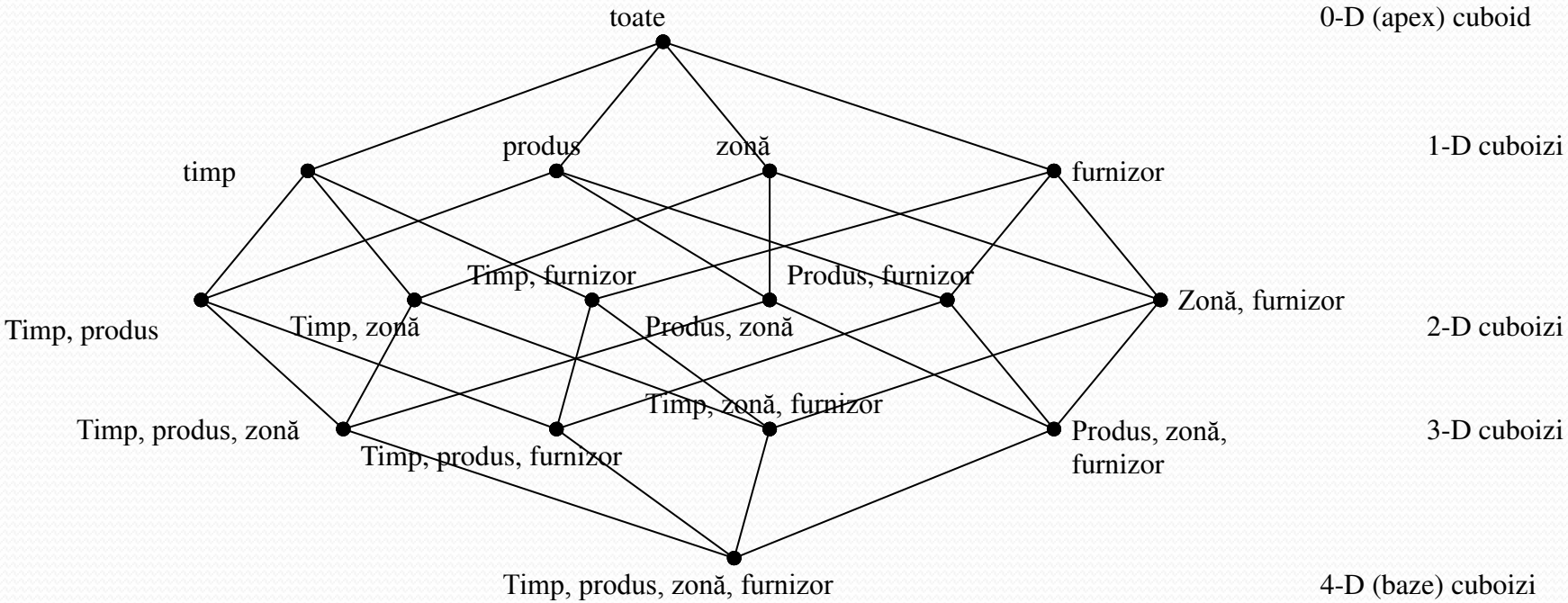
Datele 3D sunt reprezentate ca serii de tabele 2D.



Cub 3D reprezentând datele .

DD-matricea cuboid-ului

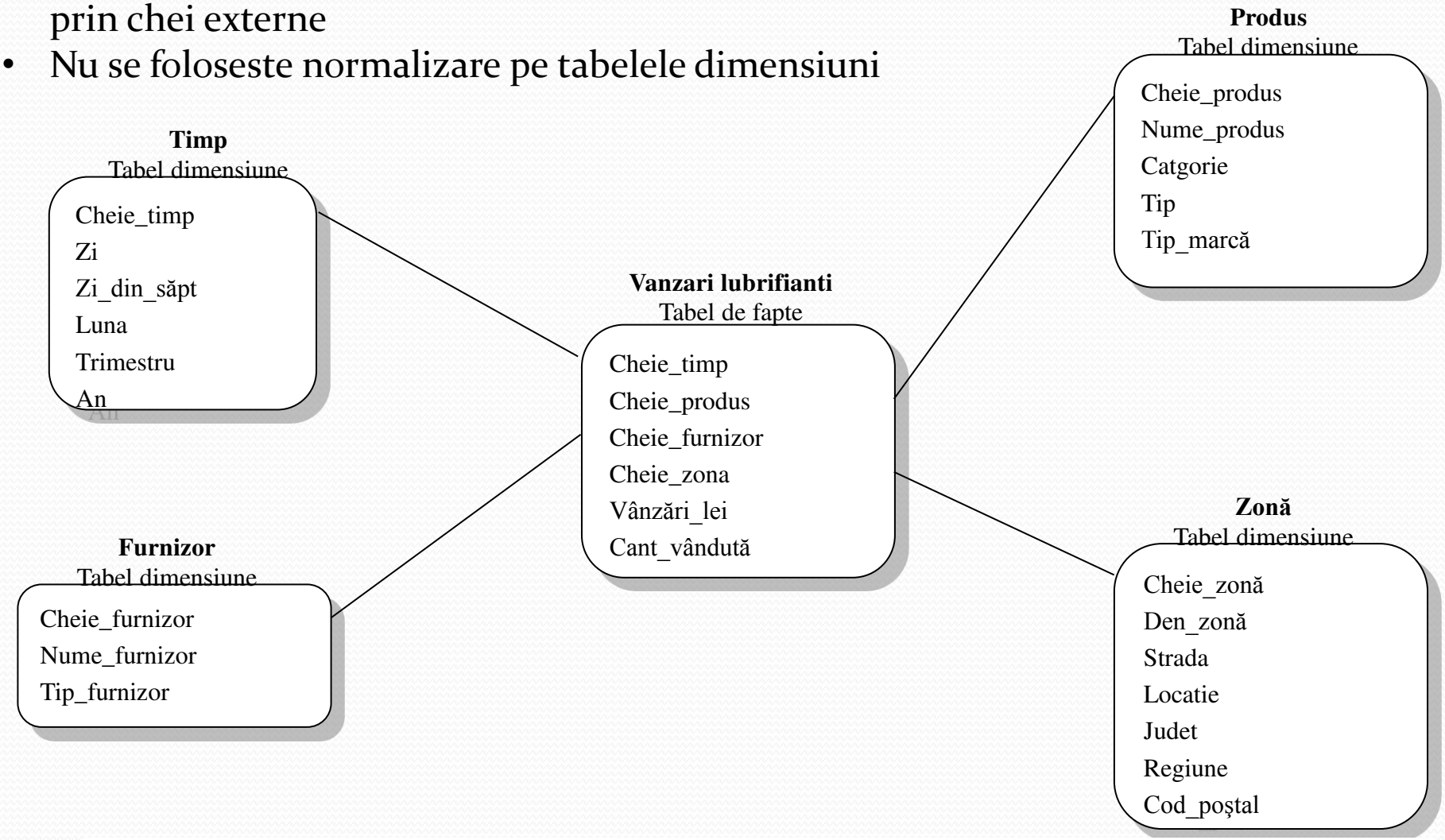
- În literatura data warehouse cubul de date este denumit cuboid.



Matricea cuboidului

DD-schema stea

- Tabela de fapte (masuri) se conecteaza la tabelele dimensiune prin chei externe
- Nu se foloseste normalizare pe tabelele dimensiuni



DD-schema *fulg de zapada*

- **Modelul fulg de zăpadă** este o variantă a modelului stea în care **o parte din tabelele dimensiune sunt normalizate**, iar datele sunt împărțite în tabele suplimentare.

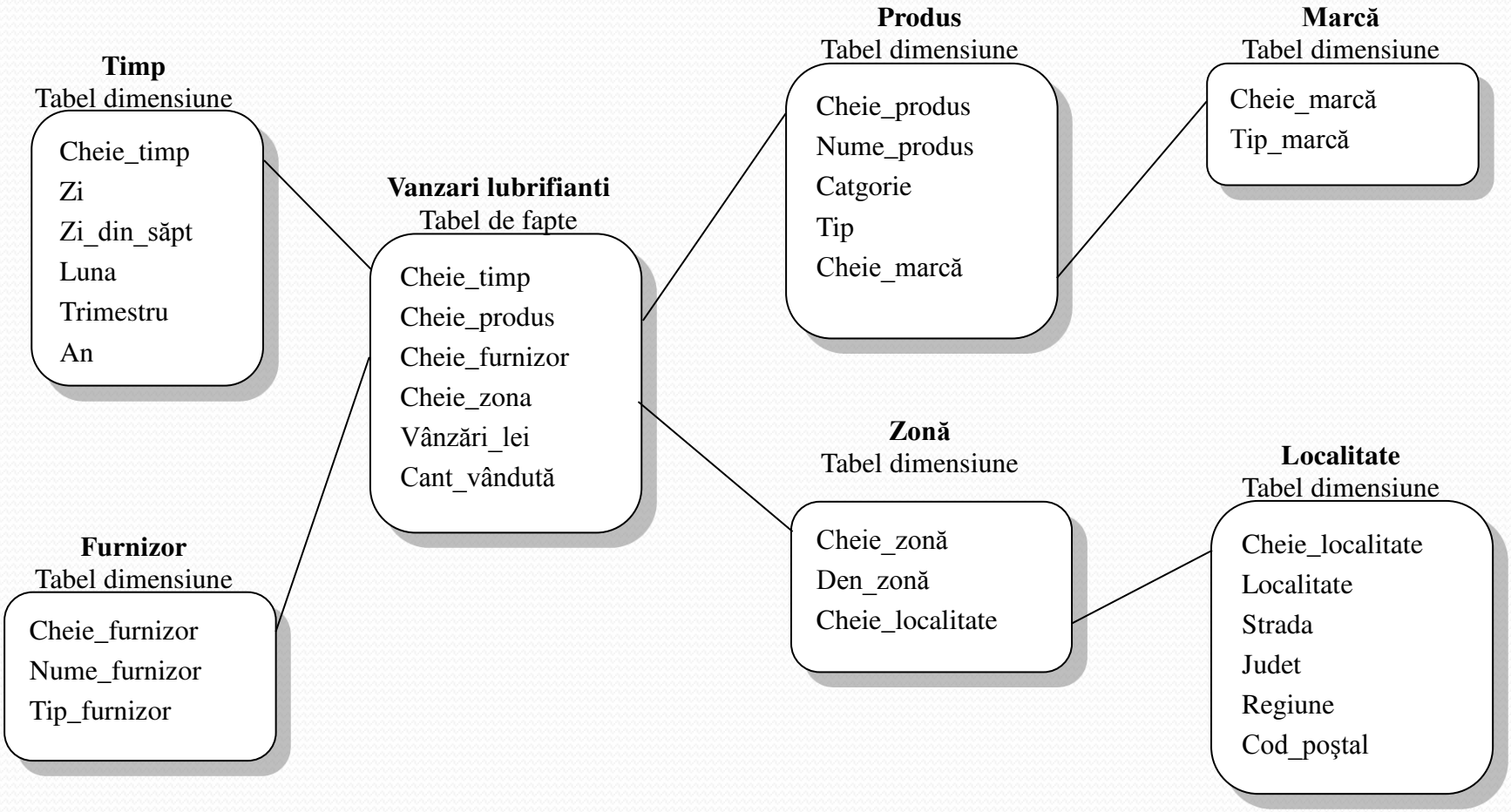
- Rezultă o schemă reprezentată într-un grafic similar unui fulg de zăpadă.

- Diferența majoră între modelul fulg de zăpadă și modelul stea este că tabelele dimensiune din modelul fulg de zăpadă pot fi păstrate în forma normalizată, ceea ce determină o **redundanță redusă.**

DD-shema *fulg de zapada*

- Asemenea tabele sunt ușor de întreținut și astfel se economisește spațiu de stocare, deoarece un tabel dimensiune mare poate deveni enorm când structura dimensională este inclusă în coloane. Totuși această economie de spațiu este neglijabilă în comparație cu volumul foarte mare de date din tabelul de fapte.
- Mai mult, structura fulg de zăpadă poate reduce eficacitatea “browsing-ului” când mai multe “join-uri” trebuie executate la o interogare. De aceea, schema fulg de zăpadă este mai puțin răspândită față de schema stea în proiectarea depozitelor de date

DD-shema fulg de zapada-vanzari lubrifianti



Schema fulg de zăpadă a unui depozit de date pentru vânzări lubrifianţi