

Baze de date

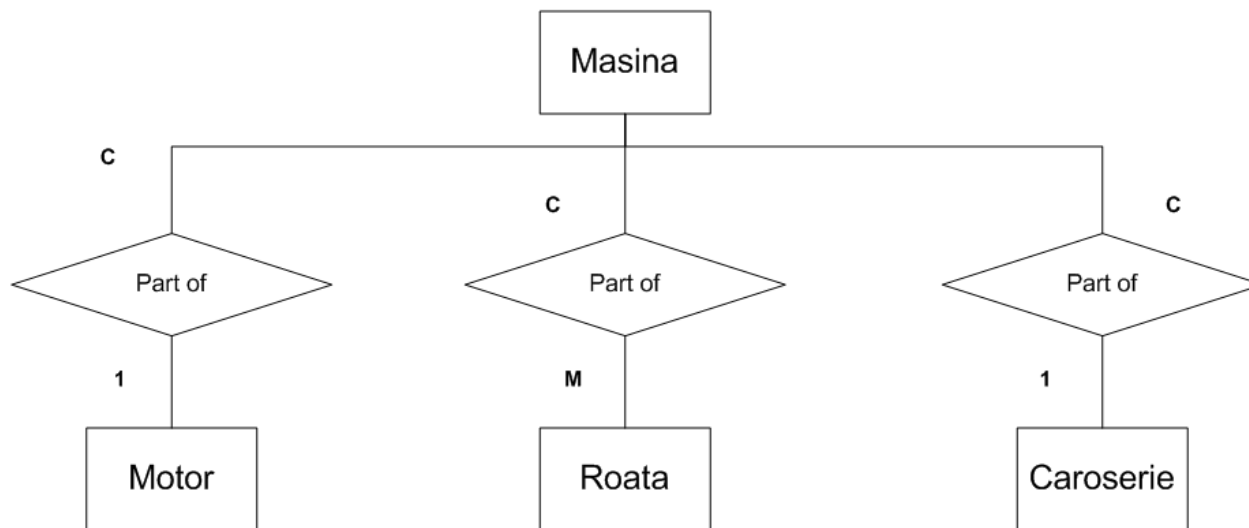
Universitatea Transilvania din Braşov

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

Asocieri speciale

□ Agregarea

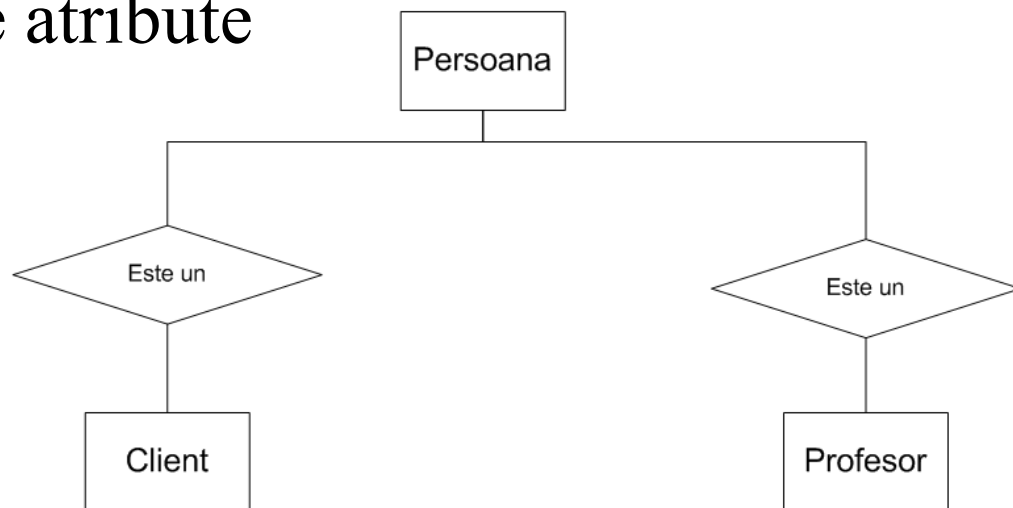
- Este descrisă ca supraordonare sau subordonare
- Adesea are forma “is part of”



Asocieri speciale

□ Generalizarea

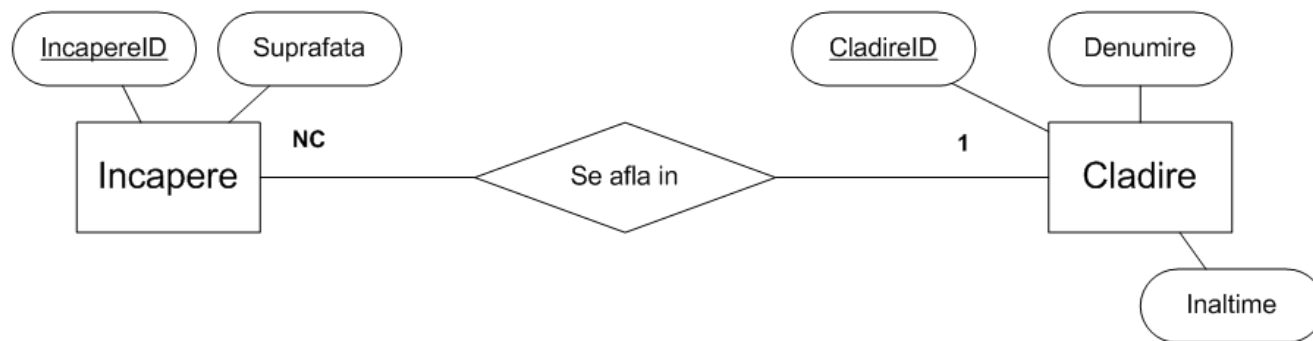
- Descrie o ierarhie
- Adesea întâlnită ca “este o/este un” (engl. “is a”)
- Atributele sunt moștenite
- Se pot adăuga alte atribute



Asocieri speciale

□ Dependenta slaba (entitate slaba)

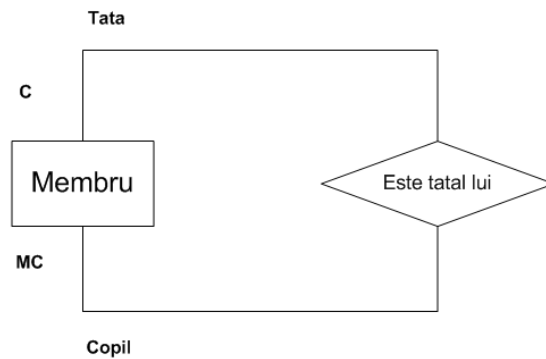
- O entitate nu poate exista fără existența unei alte entități



- Orice clădire are una sau mai multe încăperi
- Fiecare încăpere aparține unei clădiri și nu poate exista fără aceasta

Asocieri recursive

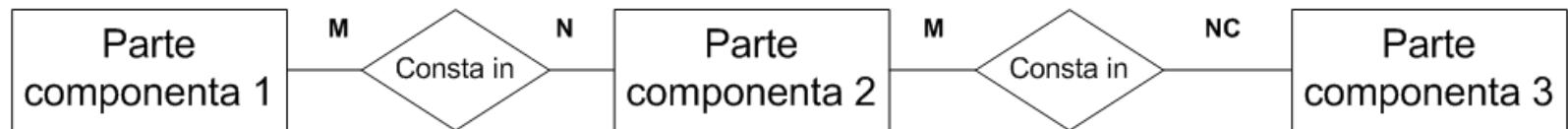
- Tip de entitate care este în relație cu ea însăși



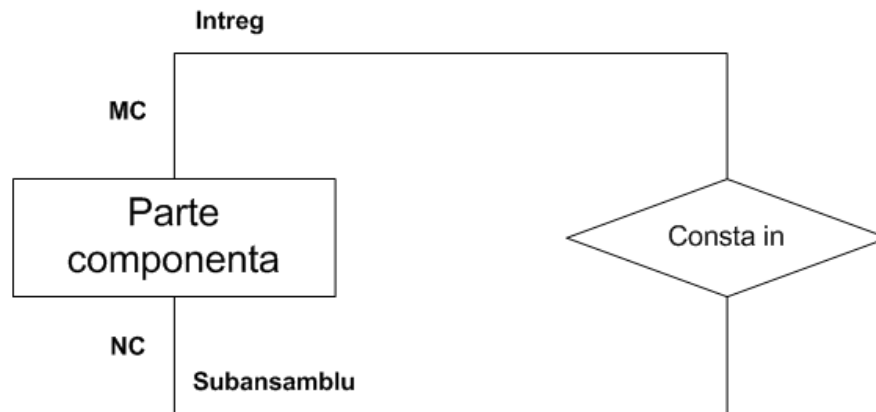
- Fiecare membru al clubului este tatăl a 0 sau mai mulți copii
- Pentru fiecare membru al clubului tatăl poate aparține sau nu clubului
- Diagrama conține și numele rolurilor

Asocieri recursive

- ❑ Relațiile recursive sunt uneori necesare (denumite și reflexive)
- ❑ Exemplu: o parte componenta este alcătuită din alte părți componente care la rândul lor sunt alcătuite din alte părți componente,...
- ❑ Chiar dacă se cunoaște nivelul de recursivitate este incorect să se modeleze folosind niveluri:



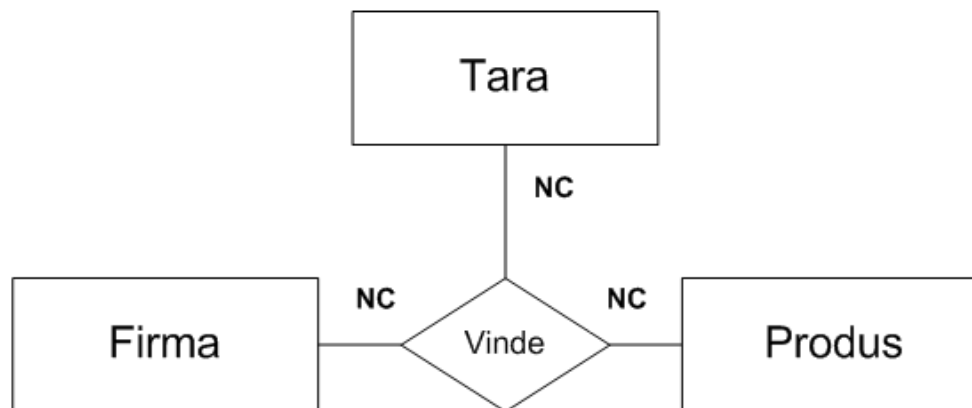
- ❑ Corect este să se modeleze folosind relații recursive:



❑ -

Asociere cu grad mai mare ca 2

- Un exemplu de relație ternară: trebuie salvate informații despre firme, produsele acestora și țările în care ele exporta produsele, de asemenea nu orice produs al unei firme nu este vândut în orice țară.
- Relațiile cu grad mai mare ca 2 sunt rare și trebuie înlocuite cu relații de grad 2.
- Cardinalitatea în acest caz semnifică ca fiecare țară este în relație cu oricâte perechi (firma, produs)



Modelul relațional

- ❑ Modelul relațional se bazează pe conceptul matematic de **relație**
- ❑ Aceasta este reprezentată fizic sub formă **tabelului**
- ❑ Bazele acestei tratări au fost puse de Codd, care a utilizat terminologia și conceptele din teoria mulțimilor și logica predicativă

Structura relațională a datelor

□ Relație

- *O relație este **un tabel** cu coloane și rânduri*
- Un SGBD relațional necesită ca baza de date să fie percepută de către utilizator doar sub formă de tabele
- Această percepție se aplică numai **structurii logice** a bazei de date – adică nivelurilor externe și conceptuale ale arhitecturii ANSI-SPARC
- Nu se aplică și structuri fizice a bazei de date, care poate fi implementată utilizând o varietate de structuri de stocare

Structura relațională a datelor

□ Atribut

- *Un atribut este o coloană a unei relații, cu o anumită denumire*
- O relație este reprezentată de un tabel bidimensional, în care rândurile acestuia corespund înregistrărilor individuale, iar coloanele corespund atributelor
- Atributele pot apărea în orice ordine, relația rămânând neschimbată

Structura relațională a datelor

□ Domeniu

- *Un domeniu este mulțimea de valori permise pentru unul sau mai multe attribute*
- Domeniile constituie o caracteristică extrem de puternică a modelului relațional
- Fiecare atribut dintr-o bază de date relațională este definit pe un domeniu
- Domeniile pot fi diferite pentru fiecare atribut, sau două sau mai multe attribute pot fi definite pe un același domeniu

Structura relațională a datelor

- Conceptul de domeniu este important deoarece permite utilizatorului definirea sensului și a sursei de valori pe care le poate lua atributul
- Ca rezultat, sistemului îi sunt disponibile mai multe informații și la executarea unei *operații relaționale*, pot fi evitate operațiile incorecte semantic
 - Exemplu
Nu are sens compararea unui număr de stradă cu un număr de telefon, cu toate că pentru ambele atribute domeniile de definiție sunt șiruri de caractere

Structura relațională a datelor

□ **Tuplu**

■ *Un tuplu este **un rând** dintr-o relație*

- Elementele unei relații sunt rândurile sau tuplurile dintr-un tabel
- Tuplurile pot apărea în orice ordine, relația va rămâne aceeași
- Structura unei relații, împreună cu specificarea domeniilor și a oricăror alte restricții asupra valorilor posibile este denumită uneori **intensitatea** acesteia
- Tuplurile sunt denumite **extensia** sau **starea** unei relații, care se modifică în timp

Structura relațională a datelor

□ Grad

- *Gradul unei relații reprezintă **numărul de attribute** pe care le conține aceasta*
- O relație cu un singur atribut are gradul întâi și este denumită relație **unară**
- O relație cu două attribute este denumită **binară**
- O relație cu trei attribute se numește **ternară**
- O relație cu n attribute se numește **n -ară**

Structura relațională a datelor

□ **Cardinalitate**

- *Cardinalitatea unei relații reprezintă **numărul de tupluri** conținute de aceasta*

- Cardinalitatea se modifică prin adăugarea sau ștergerea unor tupluri



Structura relațională a datelor

□ **Bază de date relațională**

- *Un set de relații normalizate*

□ O bază de date relațională constă în relații, care sunt structurate adecvat

- Această structurare este denumită *normalizare*

Terminologie

Filiale

| NrFil | Adresa | Orașul | CodPostal | Telefon | Fax |
|--------------|--------------------|------------------|------------------|----------------|---------------|
| F3 | Rozelor 25 | Timișoara | 1700 | 121212 | 121212 |
| F4 | Stejeriș 19 | Brașov | 2200 | 232323 | 232323 |
| F5 | Eroilor 35 | Timișoara | 1700 | 434343 | 434343 |
| F6 | Unirii 10 | Focșani | 1500 | 454545 | 454545 |

cardinalitate

grad

Cheie primară

Relație

Atribute

Cheie străină

Angajați

| NrMarca | Nume | Prenume | Adresa | Orașul | Funcția | Salariul | NrFil |
|----------------|-----------------|-----------------|---------------------|------------------|------------------|-----------------|--------------|
| 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 |

Relații matematice

- **Produsul cartezian** a două mulțimi D_1 și D_2 , scris sub forma $D_1 \times D_2$
 - reprezintă mulțimea tuturor perechilor ordonate astfel încât primul element să fie membru al mulțimii D_1 , iar al doilea element să fie membru al mulțimii D_2

Relații matematice

■ Exemplu

- Presupunem că avem două mulțimi, D_1 și D_2 , unde

$$D_1 = \{2, 4\} \text{ și } D_2 = \{1, 3, 5\}$$

- Produsul cartezian al acestor două mulțimi este

$$D_1 \times D_2 = \{(2, 1), (2, 3), (2, 5), (4, 1), (4, 3), (4, 5)\}$$

Relații matematice

□ Orice submulțime a produsului cartezian este o relație

■ Exemplu

□ Se poate realiza o relație R ca submulțime a produsului cartezian $D_1 \times D_2$ al mulțimilor $D_1 = \{2,4\}$ și $D_2 = \{1,3,5\}$ astfel încât

$$R = \{(2,1), (4,1)\}$$

Relații matematice

- Putem extinde noțiunea de relație la trei mulțimi
- Fie trei mulțimi, D_1, D_2, D_3 . Produsul cartezian $D_1 \times D_2 \times D_3$ al acestor trei mulțimi reprezintă mulțimea tuturor tripletelor ordonate, în care
 - primul element îi aparține lui D_1
 - al aparține mulțimii D_2
 - al treilea element aparține lui D_3
- Orice submulțime a acestui produs cartezian reprezintă o relație

Relații matematice

- Putem extinde noțiunea de produs cartezian la n mulțimi D_1, D_2, \dots, D_n
- Produsul cartezian al acestora este definit ca:
$$D_1 \times D_2 \times \dots \times D_n = \{(d_1, d_2, \dots, d_n) \mid d_1 \in D_1, d_2 \in D_2, \dots, d_n \in D_n\}$$
- Se mai poate scrie sub forma:
$$\bigtimes_{i=1}^n D_i$$
- Orice submulțime a acestui produs cartezian reprezintă o relație a celor n mulțimi
- În definirea acestor relații trebuie specificate mulțimile sau **domeniile** din care se aleg valori

Relații în bazele de date

□ Schema de relație

- *O denumire a relației, urmată de un set de perechi de attribute și denumiri de domenii*

- Fie attributele A_1, A_2, \dots, A_n , cu domeniile D_1, D_2, \dots, D_n .
Atunci mulțimea

$$\{A_1:D_1, A_2:D_2, \dots A_n:D_n\}$$

reprezintă schema de relație

- O relație R , definită de schema de relație S , este un set de corespondențe între denumirile atributelor și domeniile corespunzătoare acestora

Relații în bazele de date

- Prin urmare, relația R este o mulțime de n tupluri

$$\{A_1:D_1, A_2:D_2, \dots A_n:D_n\}$$

astfel încât $d_1 \in D_1, d_2 \in D_2, \dots d_n \in D_n$

- Fiecare element din n -tuplu este format dintr-un atribut și o valoare a acestuia
- În acest mod putem considera o relație din modelul relațional ca pe o submulțime a produsului cartezian al atributelor și a domeniilor

Proprietățile relațiilor

- O relație are următoarele caracteristici
 - are o **denumire**, diferită de toate celelalte denumiri de relații
 - fiecare celulă a relației conține o valoare singulară
 - fiecare atribut are o valoare distinctă

Proprietățile relațiilor

- ❑ Caracteristici (continuare)
 - toate valorile unui atribut aparțin aceluiași domeniu
 - ordinea atributelor nu are nici o importanță
 - **fiecare tuplu este distinct**, nu există dubluri ale tuplurilor
 - teoretic, ordinea tuplurilor nu are nici o importanță, dar practic, ordinea poate afecta eficiența accesării tuplurilor

Proprietățile relațiilor

■ Exemplu

- Din moment ce fiecare celulă trebuie să conțină doar o singură valoare, stocarea a două numere de telefon pentru o singură filială din relația *Filiale* care a fost prezentată în exemplul anterior este ilegală
- Cu alte cuvinte, relația nu conține grupuri repetitive
- O relație care nu conține grupuri repetitive se spune că este **normalizată** sau în **prima formă normală**

Proprietățile relațiilor

- Majoritatea proprietăților de mai sus provin din proprietățile relațiilor matematice:
 - din moment ce relația este o mulțime, ordinea elementelor sale nu are nici o semnificație
 - într-o mulțime nu se repetă nici un element
- Totuși, într-o relație matematică, ordinea elementelor dintr-un tuplu are importanță
 - Exemplu
 - Perechea ordonată $(1, 2)$ este diferită de perechea ordonată $(2, 1)$
- Acest fapt nu este valabil și pentru relațiile din modelul relațional, care necesită în mod special ca ordinea atributelor să nu prezinte importanță

Chei relaționale

- Pentru identificarea unică a unui tuplu dintr-o relație, nu sunt necesare valorile tuturor atributelor sale
- Sunt suficiente doar valorile unui subset al atributelor
- **Cheia** este un subset al atributelor unei relații care satisface proprietățile de:
 - **identificare unică** - fiecare tuplu al relației este identificat unic de valorile atributelor care compun cheia
 - **neredundanță** - subsetul de attribute este minimal, adică eliminarea oricărui atribut din subset duce la pierderea primei proprietăți

Chei relaționale

- ❑ Problema găsirii unei chei se reduce la determinarea setului minimal de attribute care satisface proprietatea de identificare unică
- ❑ Orice atribut al unei relații care face parte din cel puțin o cheie se numește **atribut prim**
- ❑ Toate celelalte attribute ale relației sunt **neprime**
- ❑ Într-o relație pot exista mai multe chei - **chei candidat**
- ❑ Pentru fiecare relație se desemnează dintre acestea o cheie privilegiată - **cheie primară**

Chei relaționale

□ Cheie primară

- *Cheia candidat care este selectată pentru a identifica în mod unic tuplurile din cadrul unei relații*
- Statutul de cheie primară al unei chei candidat este stabilit de utilizator
- Cheile candidat care nu sunt selectate drept chei primare se numesc **chei alternative**
- La selectarea unei chei primare din mulțimea cheilor candidate se va tine seama de necesitatea ca numărul atributelor cheii primare să fie cât mai mic posibil

Chei relaționale

- O **cheie străină** pentru relația R2 este un subset de attribute din R2 astfel încât
 - Există o relație de bază R1 care are o cheie candidat
 - Fiecare valoare a cheii străine din relația R2 se regăsește între valorile cheii candidat din R1

Ilustrarea unei relații

| R | A | B | C | D |
|---|-----|---|-----|-----|
| | xyz | 2 | blo | 4.6 |
| | dfg | 5 | bli | 2.4 |
| | ⋮ | ⋮ | ⋮ | ⋮ |
| | ggg | 7 | bum | 4.2 |

- In acest exemplu este reprezentata relația
$$R \subseteq A \times B \times C \times D$$
- Din descrierea formală a unui tabel ca relație se observă următoarele:
 - Toate valorile unei coloane au același tip
 - Toate liniile sunt diferite (relațiile sunt mulțimi)
 - Ordinea liniilor este oarecare
 - Semnificația unei coloane este descrisă printr-un nume (domeniul de valori)

Transformarea modelului ER în relații

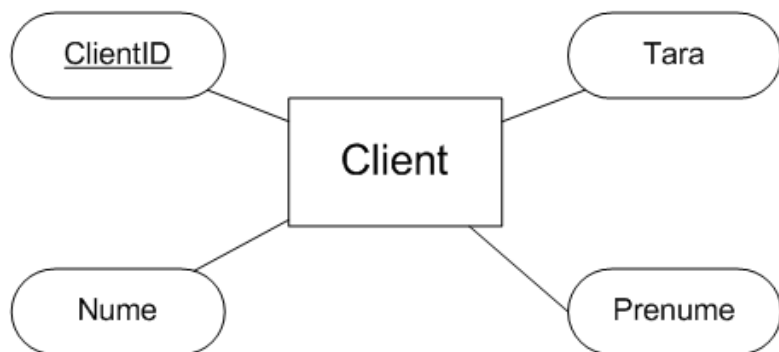
- ❑ Modelele ER pot fi transformate în relații fără pierderea de informații
- ❑ Se aplica reguli diferite în funcție de cardinalitatea relațiilor
- ❑ Este importanta manipularea cheilor primare și cheilor străine

Scopul transformării

- ❑ La completarea tabelelor cu date trebuie evitate datele redundante
- ❑ Dacă problema modelata nu are nevoie de valori NULL atunci acestea se vor elimina
- ❑ Se încearcă crearea unui număr minim de tabele ținând cont de regulile de mai sus.

Transformarea tipurilor de entități

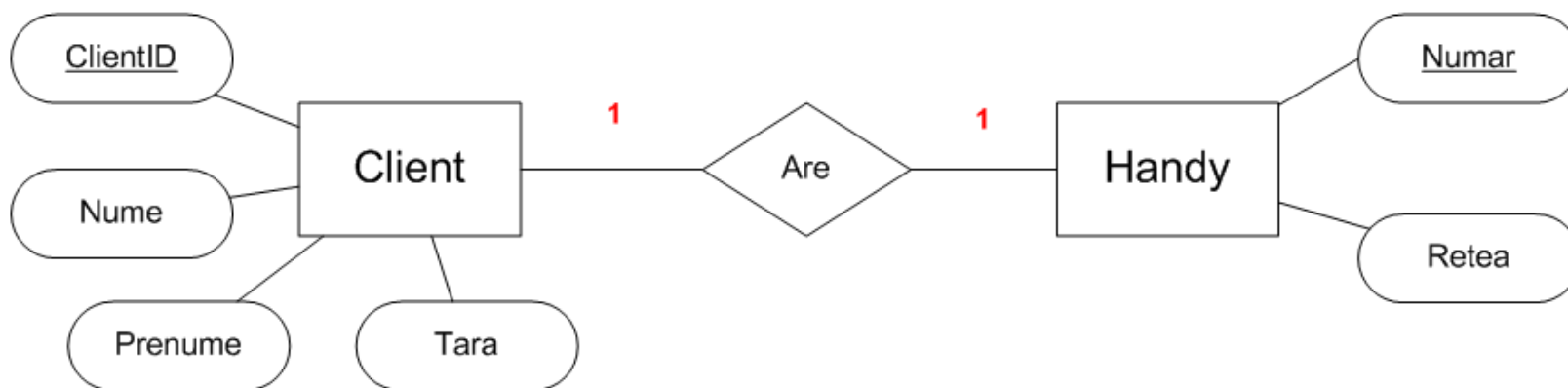
- ❑ Tipurile de entități devin tabele
- ❑ Atributele devin coloane
- ❑ Fiecare entitate devine linie sau înregistrare
- ❑ Client(ClientID, Nume, Prenume, Tara)



| <u>ClientID</u> | Nume | Prenume | Tara |
|-----------------|------------|---------|----------|
| 1 | Mustermann | Max | Germania |
| 2 | Doe | John | SUA |

Transformarea relațiilor 1:1

- Informațiile sunt asamblate într-un singur tabel

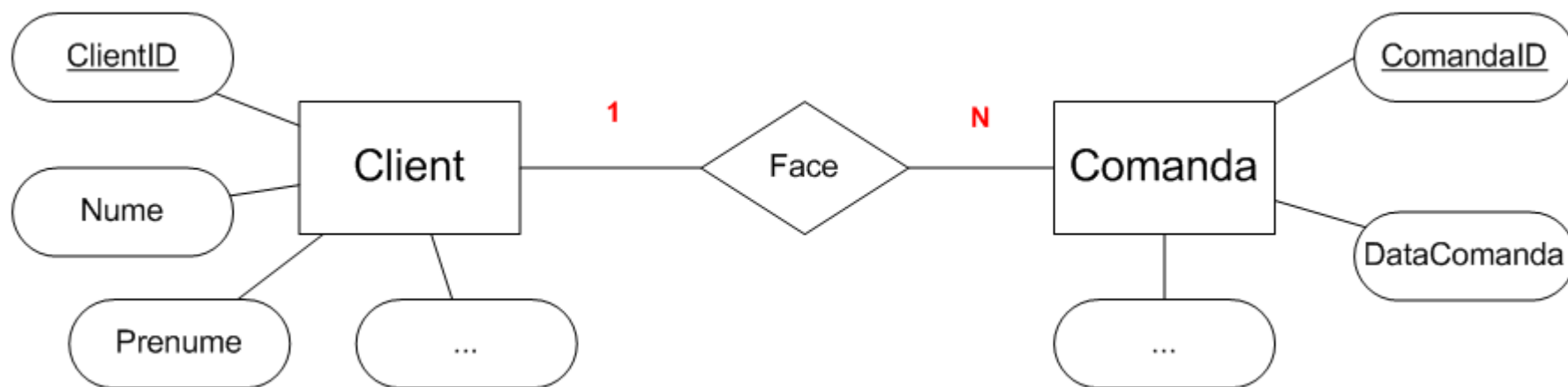


- Client(ClientID, Nume, Prenume, Tara, Număr, Rețea)

| <u>ClientID</u> | Nume | Prenume | Tara | | Numar |
|-----------------|------------|---------|----------|-----|-------------|
| 1 | Mustermann | Max | Germania | ... | 0162/234123 |
| 2 | Doe | John | SUA | ... | 555/1231456 |

Transformarea relațiilor 1:N

- Sunt necesare doua tabele
 - Tabelul Client
 - Tabelul Comanda – conține cheia primara a tabelului supraordonat (cel marcat cu “1”) care va fi denumit cheie străină.



Transformarea relațiilor 1:N

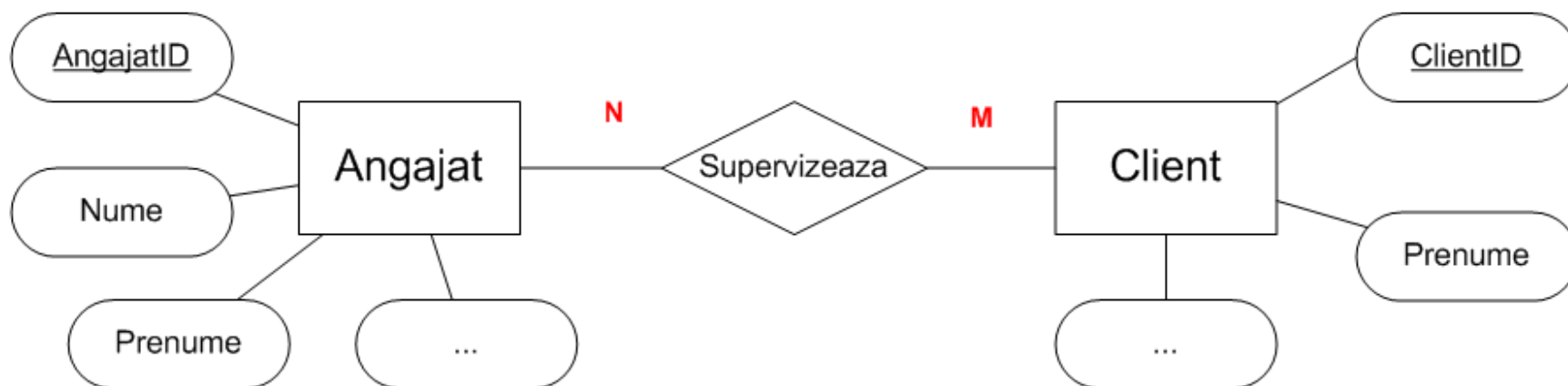
| <u>ClientID</u> | Nume | Prenume | Tara | | Numar |
|-----------------|------------|---------|----------|-----|-------------|
| 1 | Mustermann | Max | Germania | ... | 0162/234123 |
| 2 | Doe | John | SUA | ... | 555/1231456 |
| ... | | | | | |



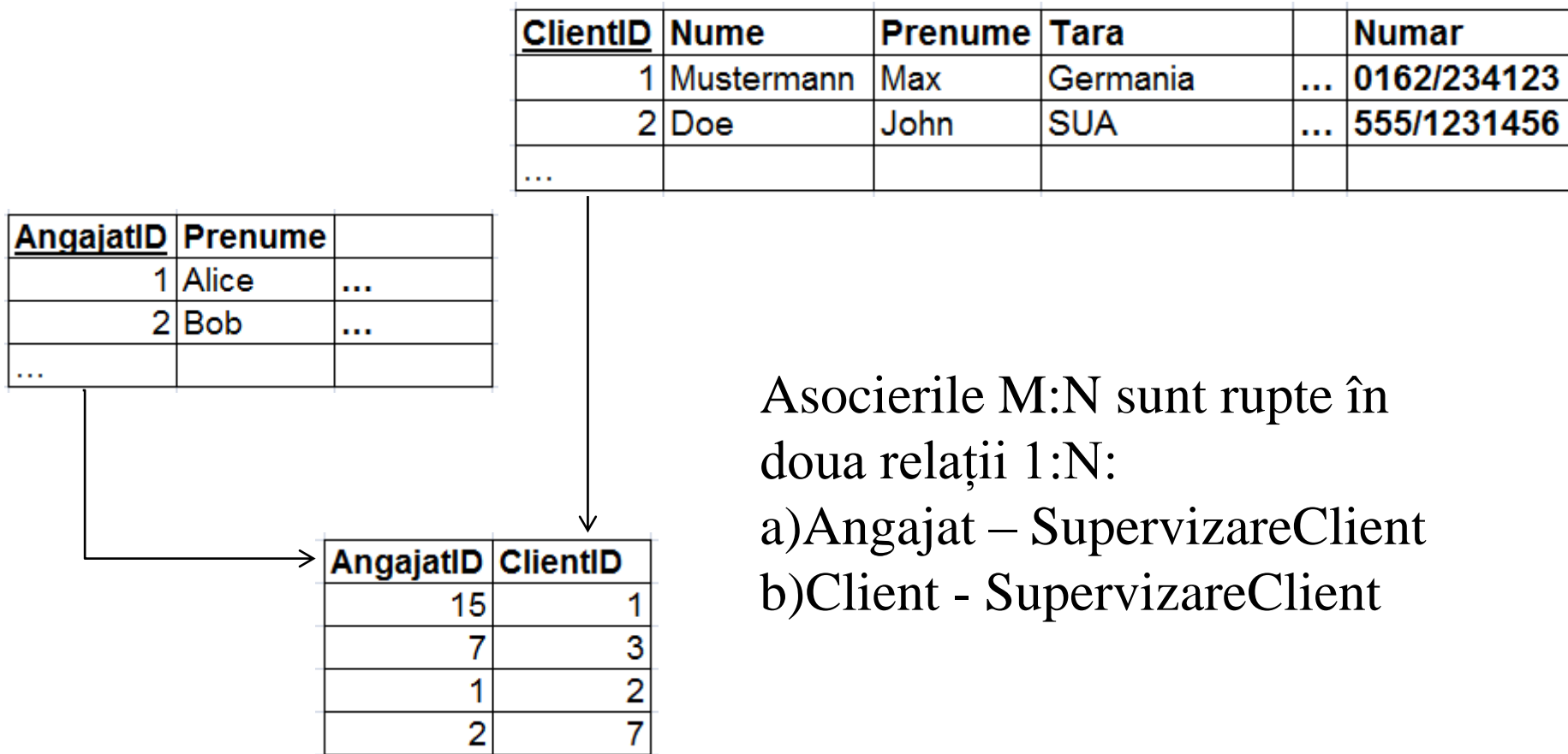
| <u>ComandaID</u> | ClientID | DataComanda | ... |
|------------------|----------|-------------|-----|
| 1 | 1 | 22.01.2016 | |
| 2 | 1 | 30.03.2016 | |
| 3 | 2 | 15.09.2016 | |
| 4 | 5 | 05.11.2016 | |
| ... | | | |

Transformarea relațiilor M:N

- Sunt necesare trei tabele
 - Tabelul Angajat
 - Tabelul Client
 - Tabelul de legătură SupervizareClient care conține cheile primare din celelalte tabele (si attributele asocierii daca este cazul)
 - Cheia primara a acestui tabel (SupervizareClient) poate fi formata din cele doua chei străine

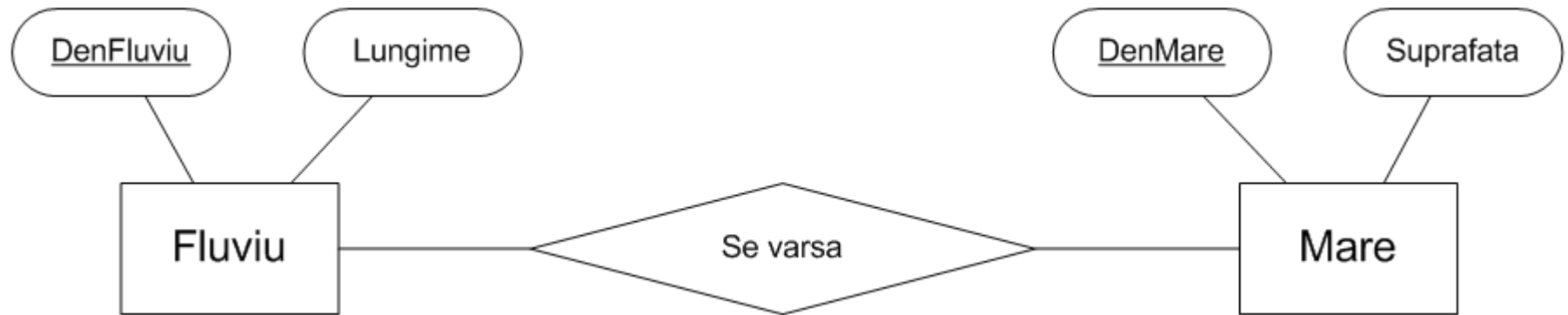


Transformarea relațiilor M:N



Asocierile M:N sunt rupte în
doua relații 1:N:
a) Angajat – SupervizareClient
b) Client - SupervizareClient

Transformarea relațiilor ce au cardinalitate “C”



- Scop principal: evitare valori nule daca este posibil

Fluviu

| <u>DenFluviu</u> | Lungime | DenMare |
|------------------|---------|--------------------|
| ... | | |

Mare

| <u>DenMare</u> | Suprafata | DenFluviu |
|----------------|-----------|----------------------|
| ... | | |

Afluent

| <u>DenFluviu</u> | DenMare |
|------------------|---------|
| ... | |