

Baze de date

Universitatea “Transilvania” din Brasov

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

Arhitectura client –server

- Arhitectura client - server se referă la modul în care interacționează componentele de software pentru a forma un sistem care constă din
 - proces client - care cere resurse
 - proces server - care oferă resurse
- Nu există nici o cerință ca atât clientul, cât și serverul să se afle pe același calculator
- În practică, se obișnuiește să se plaseze serverul pe un server din rețeaua locală și clienții pe alte mașini

Arhitectura client –server

- ❑ Clientul administrează interfața cu utilizatorul și logica aplicației, acționând ca o stație de lucru, pe care sunt executate aplicațiile bazei de date
- ❑ Clientul preia cererea utilizatorului, verifică sintaxa și generează cererea pentru baza de date în limbajul SQL sau în alt limbaj de baze de date, adecvat logicii aplicației
- ❑ Apoi transmite mesajul serverului, așteaptă un răspuns și îl formatează pentru utilizatorul final
- ❑ Serverul acceptă și procesează cererea pentru baza de date, după care transmite rezultatul înapoi clientului

Arhitectura client –server

- Procesarea implică:
 - verificarea autorizării
 - asigurarea integrității
 - menținerea catalogului de sistem
 - execuția proceselor de interogare și reactualizare
 - asigurarea controlului simultaneității și reconstituirii

Operațiunile clientului și serverului

<i>Client</i>	<i>Server</i>
Administrează interfața cu utilizatorul	Primește și procesează cerințele clienților pentru baza de date
Acceptă și verifică sintaxa intrărilor utilizatorilor	Verifică autorizarea
Procesează aplicațiile	Asigură respectarea contrângerilor de integritate
Generează cerințele pentru baza de date și le transmite serverului	Efectuează procesarea interogare/reactualizare și transmite clientului răspunsul
Transmite răspunsul înapoi la utilizator	Întreține catalogul sistem
	Oferă acces simultan la baza de date
	Oferă controlul reconstituirii

Arhitectura client –server

□ Avantaje

- permite un **acces mai larg la bazele de date** existente
- are performanțe crescute - dacă serverul și clienții se află pe calculatoare diferite, atunci diferite unități CPU **pot procesa aplicații în paralel**
- **configurarea serverului** este mai ușor de efectuat dacă singura sa sarcină este de a efectua prelucrarea bazei de date

Arhitectura client –server

□ Avantaje

■ **reduce costurile dispozitivelor hardware**

- numai serverul necesită o capacitate de stocare și o putere de prelucrare suficiente pentru a stoca și gestiona baza de date

■ **reduce costurile comunicațiilor**

- aplicațiile execută o parte din operații la client, care trimite prin rețea numai cererea de acces la baza de date, ceea ce face ca pe rețea să circule mai puține date

Arhitectura client –server

□ Avantaje

■ mărește coerența

- serverul poate trata verificările de integritate, deoarece **constrângerile trebuie definite și validate într-un singur loc**, fără să fie necesar ca fiecare program aplicație să execute propriile verificări

■ se transpune destul de natural într-o arhitectură de sisteme deschise

Arhitectura client –server

- ❑ Cu toate că arhitectura client-server poate fi utilizată pentru a oferi SGBD distribuite, totuși ea însăși nu constituie un SGBD distribuit
- ❑ În arhitectura client-server pe nivele clientul manevrează numai interfața cu utilizatorul, în timp ce nivelul intermediar manevrează logica aplicațiilor
- ❑ Al treilea nivel îl constituie serverul bazei de date
- ❑ Această arhitectură pe trei nivele s-a dovedit a fi mai convenabilă pentru unele medii cum ar fi Internet și rețelele intranet ale companiilor unde un browser Web poate fi utilizat drept client

Catalogul sistem

- ❑ Este un depozit de informații care descriu datele din baza de date, adică metadatele sau datele despre date
- ❑ Un SGBD trebuie să posede un **catalog de sistem** sau un **dicționar de date** accesibil utilizatorilor

Catalogul sistem

- ❑ Catalogul de SGBD este una din componentele de bază ale sistemului
- ❑ Volumul de date conținut și modul în care sunt utilizate informațiile variază de la sistem la sistem

Catalogul sistem

- Catalogul sistem stochează
 - denumirile, tipurile și dimensiunile articolelor de date
 - denumirile relațiilor
 - constrângerile de integritate asupra datelor
 - numele utilizatorilor autorizați care au acces la date

Catalogul sistem

- Catalogul sistem stochează
 - articolele de date pe care le poate accesa fiecare utilizator și tipurile de acces permise
 - schemele externe, conceptuale și interne și transpunerile dintre ele
 - statistica utilizării
 - frecvența tranzacțiilor
 - contorizarea numărului de accesări ale obiectelor din baza de date

Catalogul sistem

- Termenul de **dicționar de date** este adesea utilizat pentru a face referire la un sistem software mai general decât catalogul unui SGBD
- Un sistem de dicționare de date poate să fie
 - **activ** - este întotdeauna coerent cu structura bazei de date, deoarece este întreținut automat de sistem
 - **pasiv** - poate să nu fie coerent cu baza de date, deoarece schimbările sunt inițiate de utilizatori

Catalogul sistem

- ❑ Dicționarul de date este **integrat** dacă dicționarul de date este o parte a sistemului SGBD
- ❑ Un dicționar de date **autonom** are propriul său sistem SGBD specializat

Catalogul sistem

- ❑ Un dicționar de date autonom poate fi preferabil în stadiile inițiale ale proiectării, deoarece acesta întârzie cât mai mult posibil decizia asupra unui anumit SGBD al organizației
- ❑ Există dezavantajul că, o dată ce SGBD a fost selectat și baza de date implementată, este mult mai dificil să se mențină dicționarul de date autonom coerent cu baza de date
- ❑ Această problemă ar putea fi minimizată dacă ar fi posibilă transferarea dicționarului de date din proiectare în catalogul de SGBD

Catalogul sistem

Sistemul de informatii al dicționarului de resurse (IRDS)

- ❑ În multe sisteme, dicționarul de date este o componentă internă a sistemului SGBD, care stochează numai informațiile direct legate de baza de date
- ❑ Datele conținute de SGBD reprezintă, de obicei, numai o parte a cerințelor informaționale totale ale unei organizații

Catalogul sistem

Sistemul de informatii al dicționarului de resurse (IRDS)

- De regulă, există informații adiționale conținute în alte instrumente, cum ar fi CASE, instrumentele de documentare și instrumentele de configurare și administrare a proiectului
- Fiecare dintre aceste instrumente va avea propriul dicționar de date intern, care poate fi accesat de alte instrumente externe
- Nu a existat nicio modalitate generală de partajare a acestor seturi diferite de informații între diversele grupuri de utilizatori sau aplicații

Catalogul sistem

Sistemul de informatii al dicționarului de resurse (IRDS)

- ❑ Sistemul de informații al dicționarului de resurse (IRDS) standardizează interfața la dicționarele de date pentru a le face mai accesibile și cu posibilități superioare de partajare
- ❑ Un sistem IRDS este un instrument software care poate fi utilizat pentru a controla și documenta o resursă de informații a unei organizații
- ❑ El oferă o definiție pentru tabelele care conțin dicționarul de date și operațiile care pot fi utilizate pentru accesarea acestor tabele

Catalogul sistem

Sistemul de informatii al dicționarului de resurse (IRDS)

- Operațiile oferă o metodă coerentă de accesare a dicționarului de date și o modalitate de transferare a definițiilor datelor de la un dicționar la altul

- Exemplu

Informațiile stocate într-un dicționar de date de tip IRDS al unui sistem DB2 pot fi transferate la un dicționar de date de același tip al unui sistem ORACLE sau pot fi accesate de o aplicație DB1, utilizând serviciile IRDS

Catalogul sistem

Sistemul de informatii al dicționarului de resurse (IRDS)

- Principalul avantaj al sistemului IRDS îl reprezintă extensibilitatea dicționarului de date
- Dacă un utilizator al unui SGBD dorește să stocheze definițiile corespunzătoare unui nou tip de informație într-un instrument - de exemplu rapoartele de administrare a proiectelor dintr-un SGBD - atunci sistemul IRDS al SGBD poate fi extins pentru a include această informație

Catalogul sistem

Sistemul de informatii al dicționarului de resurse (IRDS)

- Standardele IRDS definesc un set de reguli referitoare la modul în care sunt stocate și accesate informațiile în dicționarul de date
- Sistemul IRDS are trei obiective
 - extensibilitatea datelor
 - integritatea datelor
 - accesul controlat la date

Catalogul sistem

Sistemul de informatii al dicționarului de resurse (IRDS)

- Sistemul IRDS se bazează pe o interfață de servicii, alcătuită dintr-un set de funcții care pot fi apelate pentru a accesa dicționarul de date
- Interfața de servicii poate fi invocată de următoarele componente ale interfeței cu utilizatorul:
 - tablou
 - limbaj de comandă
 - fișiere de export/import
 - programe aplicație

Catalogul sistem

Sistemul de informatii al dicționarului de resurse (IRDS)

- ❑ Interfața cu tabloul constă într-un set de tablouri sau ecrane, fiecare dintre ele oferind acces la un set precis de servicii
- ❑ Această interfață poate fi similară limbajului QBE (Query-by-Example) și permite utilizatorului să răsfoiască și să modifice datele din dicționar.
- ❑ Interfața cu limbajul de comandă (CLI) constă într-un set de comenzi sau instrucțiuni, care permit utilizatorului să efectueze operații asupra datelor din dicționar
- ❑ Interfața CLI poate fi invocată interactiv de la un terminal sau poate fi încorporată într-un limbaj de programare de nivel înalt

Catalogul sistem

Sistemul de informatii al dicționarului de resurse (IRDS)

- ❑ Interfața export/import generează un fișier care poate fi mutat între sistemele de tip IRDS
- ❑ Standardul definește un format general pentru interschimbarea de informații
- ❑ Standardul nu necesită ca baza de date a dicționarului să se conformeze unui anumit model de date, astfel încât interfața cu serviciile IRDS poate să conecteze sisteme SGBD eterogene

Exemplu: baza de date facultate

□ Nivel logic (de exemplu tabele)

- *Student* (*studid: int, name: string, login: string, varsta: int*)
- *Curs* (*cursid: int, cname: string, student: int*)
- *Specializare* (*spid: int, spname: string, buget: real*)
- *Preda* (*spid: int, cursid: int*)
- *Inscrisi* (*studid: int, cursid: int*)

□ Nivel fizic (intern)

- Salvarea relatiilor ca fisiere: multimi nesortate de inregistrari fizice
- Indexarea primei coloane din Student si Curs pentru a acceleara accesul la date

□ Nivel extern (View)

- **Interogare:** Cati studenti s-au inscris la fiecare curs??
- *Curs_Info* (*cursid: int, numarinscrisi: int*)



Bazele modelului entitate-asociere

- ❑ ER Entity-Relationship (entitate-relatie)
- ❑ Ce este un model?
- ❑ Ce este o entitate?
- ❑ Ce sunt relatiile?
- ❑ Ce sunt attributele?

Proiectare conceptuala (nivel logic)

- Proiectarea conceptuala a bazei de date este similara cu modelarea in stiinta si tehnica
- **Un model se construiește pentru a îmbunătăți înțelegerea și pentru a abstractiza detaliile.**
- Îmbunătățirea înțelegerii: semnificatia datelor și a relatiilor dintre ele și reprezentarea lor în structuri informationale
- Abstractizarea: neglijarea detaliilor individuale ale valorilor datelor

Modelarea datelor

- ❑ Obiectele lumii reale, importante pentru tema studiata, sunt descrise intr-un mod abstract impreuna cu relatiile dintre ele.
- ❑ Intrebari
 - Ce obiecte/entitati au un rol in modelarea problemei?
 - Ce proprietati/atribute au aceste entitati?
 - In ce relatii se afla aceste entitati?
 - Ce proprietati au aceste relatii?
- ❑ Se pune problema analizei cerintelor asupra softului astfel incat cerintele clientului sa fie transformate in cerinte asupra bazei de date

Concepte de baza ale modelului ER

- Entitate (obiect)
 - Exemplar individual, identificabil
 - Definit de proprietati
- Tip de entitate
 - Agregarea entitatilor cu aceleasi proprietati intr-o multime
 - Descrie de nume (substantiv) ca termen generic pentru toate entitatile din multime

Concepte de baza ale modelului ER

□ Atribut

- Caracteristica pentru toate entitatile unei multimi de entitati (tip de entitate)
- Are denumire cu semnificatie
- Are domeniu de valori specificat
- Descrie/identifica o caracteristica a unui tip de entitate

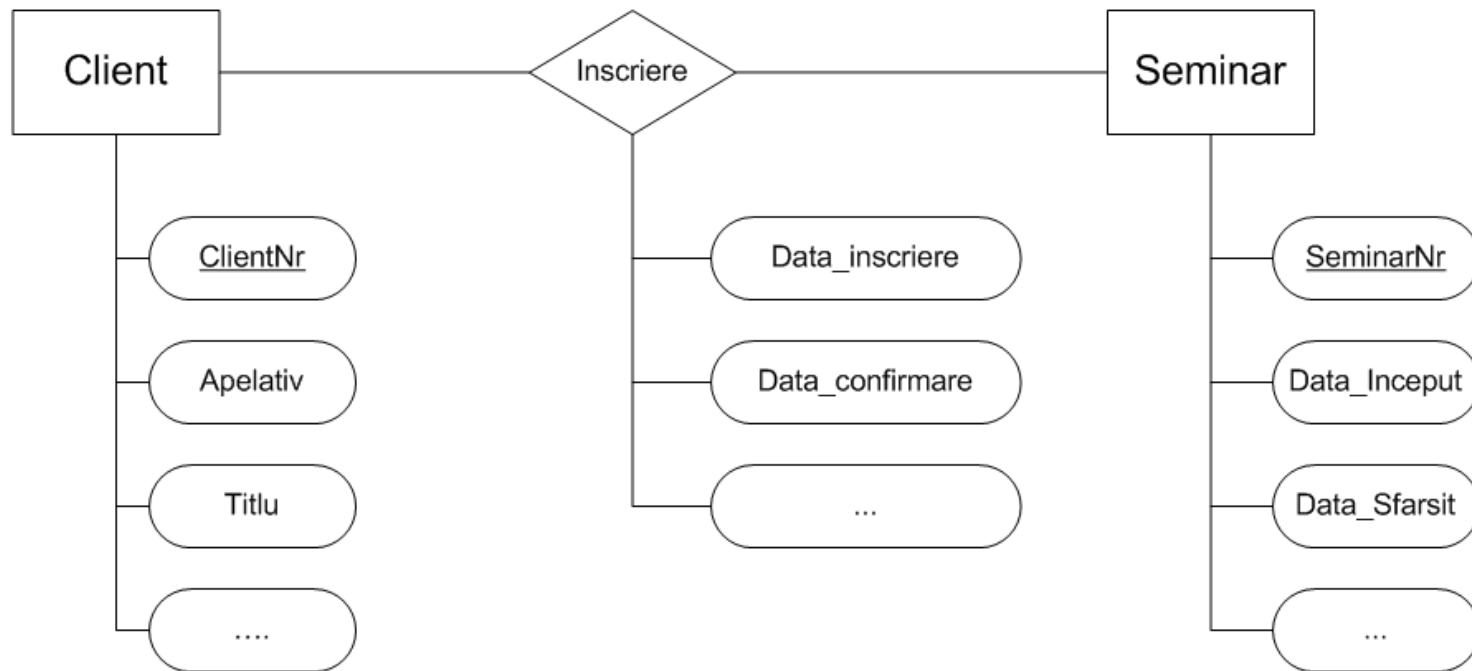
□ Cheie primara

- Combinatie de attribute care identifica in mod unic o entitate

Concepte de baza ale modelului ER

- Asociere (relatie)
 - Agrega relatiile dintre entitati
 - Ca termen de identificare este un verb care semnifica aceeasi asociere intre elementele a doua multimi de entitati (doua tipuri de entitati)
 - Poate avea attribute

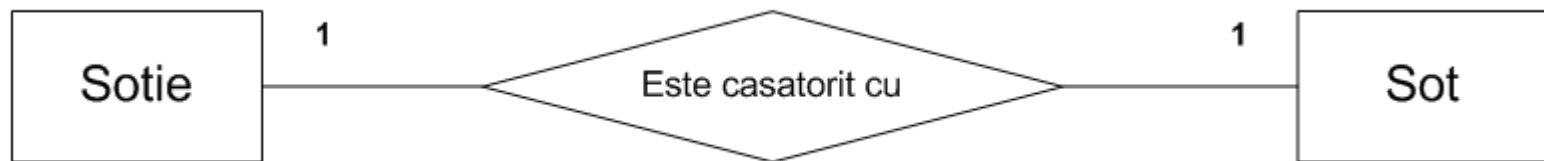
Reprezentarea grafica



- ❑ Atributul care identifica entitatea este subliniat.
- ❑ Obs: este posibila utilizarea mai mult atribute pentru identificarea unei entitati.

Cardinalitatea asocierilor

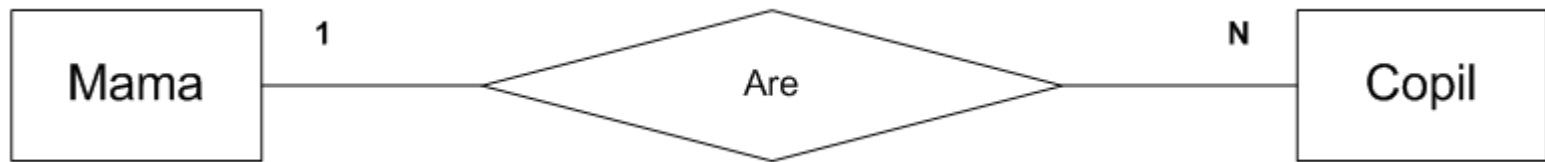
❑ Asocierea 1:1



- ❑ Fiecare sotie are exact un sot
- ❑ Fiecare sot are exact o sotie
- ❑ Corectitudinea unei diagrame depinde de cerintele asupra bazei de date

Cardinalitatea asocierilor

□ Asocierea 1:N



- Fiecare mama poate avea mai multi copii
- Fiecare mama are cel putin un copil
- Fiecare copil are exact o mama

Cardinalitatea asocierilor

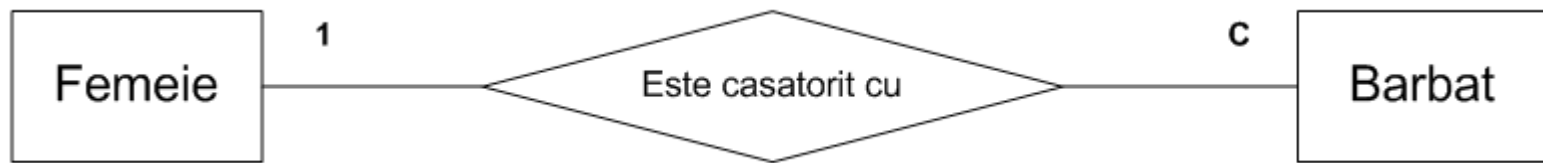
□ Asocierea M:N



- Fiecare student poate participa la mai multe cursuri
- Fiecare student participa la cel puțin un curs
- Fiecare curs poate fi urmarit de mai multi studenti
- Fiecare curs este urmarit de cel puțin un student

Cardinalitatea asocierilor

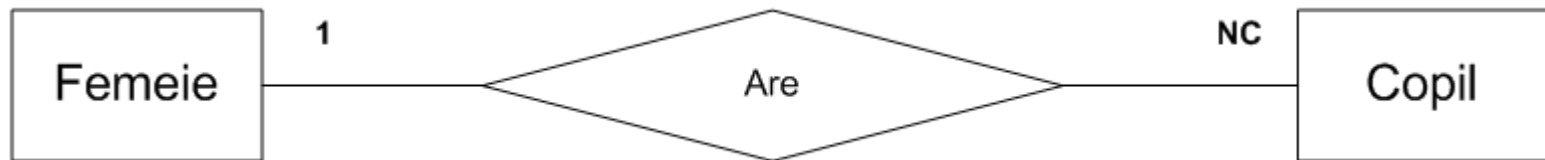
❑ Asocierea 1:C



- ❑ Fiecare femeie are fie exact un barbat fie niciunul
- ❑ Fiecare barbat are exact o femeie

Cardinalitatea asocierilor

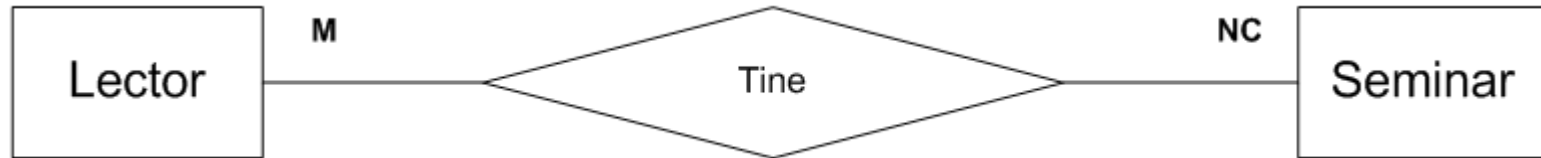
□ Asocierea 1:NC



- Fiecare femeie poate avea:
 - Niciun copil
 - Un copil
 - Mai multi copii
- Fiecare copil are exact o mama

Cardinalitatea asocierilor

□ Asocierea M:NC



□ Fiecare lector poate sa sustina:

- Niciun seminar
- Un seminar
- Mai multe seminare

□ Fiecare seminar are

- Un lector
- Mai multi lectori

Cardinalitatea asocierilor

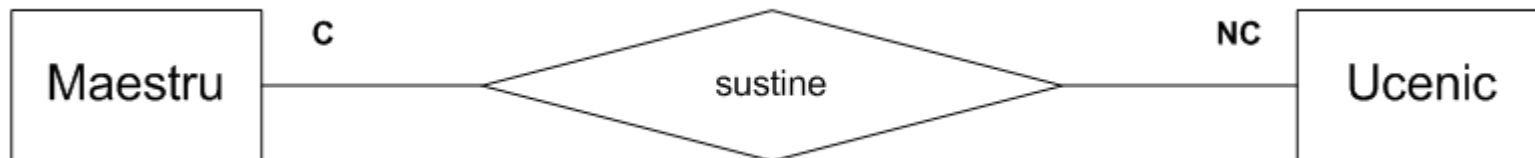
□ Asocierea MC:NC



- Fiecare articol poate sa fie:
 - Epuizat (in niciun depozit)
 - Intr-un depozit
 - In mai multe depozite
- Fiecare depozit poate sa contina
 - Nici un articol
 - Un articol
 - Mai multe articole

Cardinalitatea asocierilor

□ Asocierea C:NC



- Fiecare maestru poate sa sustina:
 - Cel putin un ucenic (altfel nu este maestru)
 - Mai multi ucenici
- Fiecare ucenic este sustinut de
 - Nici un maestru
 - Un singur maestru (altfel se creeaza confuzie)

Cardinalitatea asocierilor

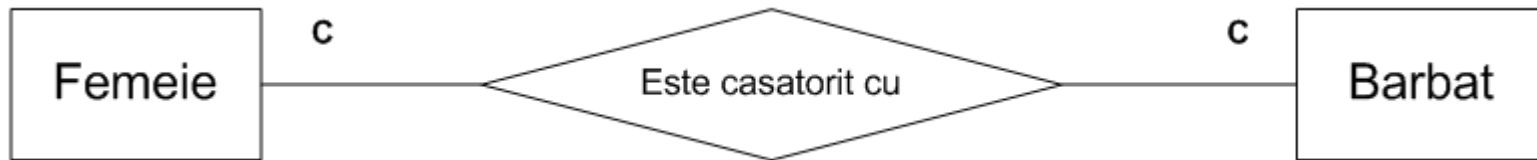
□ Asocierea NC:C



- Fiecare fluviu se varsa:
 - Exact intr-o mare
 - In nicio mare
- Fiecare mare are ca afluent
 - Niciun fluviu
 - Exact un fluviu
 - Mai multe fluvii

Cardinalitatea asocierilor

□ Asocierea C:C



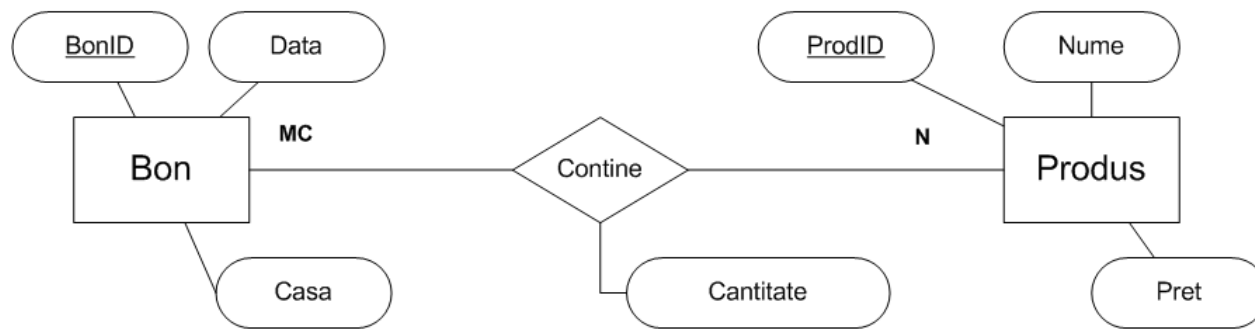
- Fiecare femeie este casatorita:
 - Cu un sot
 - Cu niciunul
- Fiecare barbat este casatorit cu
 - O sotie
 - Niciuna

Cardinalitatea asocierilor

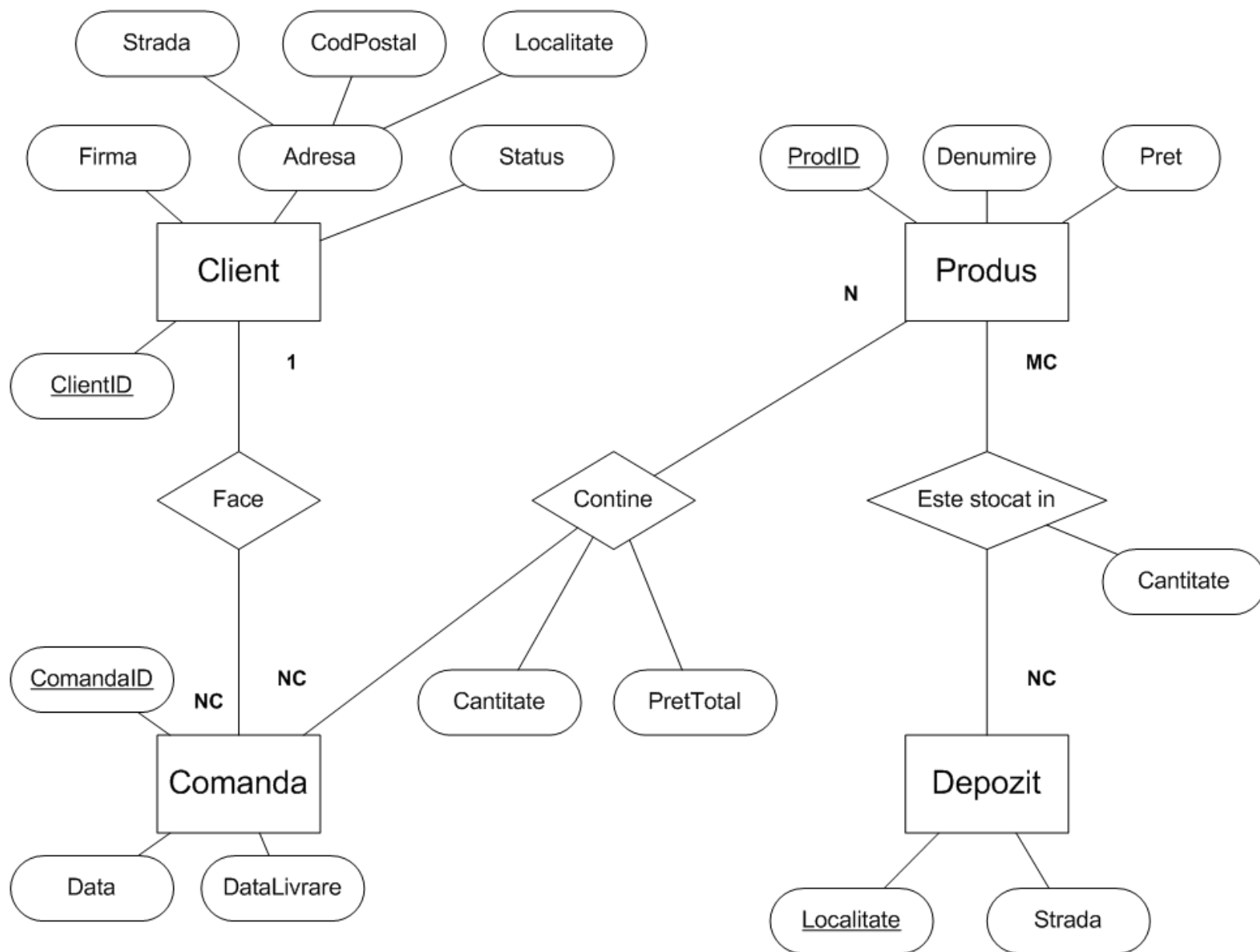
	B	Trebuie		Poate	
A		1	N	C	NC
Trebuie	1	1:1	1:N	1:C	1:NC
	M	M:1	M:N	M:C	M:NC
Poate	C	C:1	C:N	C:C	C:NC
	MC	MC:1	MC:N	MC:C	MC:NC

Asociere cu attribute

- ❑ Atributele care nu pot fi adaugate unei entitati sunt adaugate asocierilor



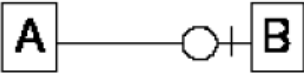
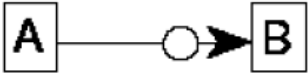
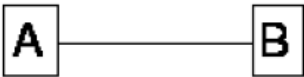

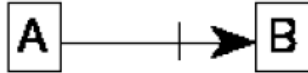

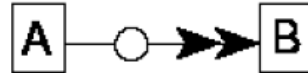
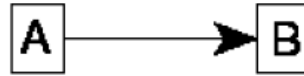

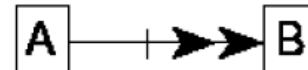
- ❑ Cantitatea nu poate fi adaugata pe bon deoarece difera pentru fiecare produs
- ❑ Cantitatea nu poate fi adaugata in produs pentru ca difera pe fiecare bon in parte



Analiza text pentru creare model

- Substantivele pot fi entitati sau attribute
- Adjectivele se refera la attribute
- Verbele reprezinta entitati (dar si attribute si asocieri) in asociere
- Probleme:
 - Sinonime: cuvinte care descriu un acelasi concept (exemplu: titlu carte sau exemplar)
 - Omonime: acelasi cuvant cu intelesuri diferite (de exemplu banca nu este un nume de entitate corect)

Notatii

C	(0,1)			
1	(1,1)			
NC	(0,n)			
N	(1,n)			

Notatia NC

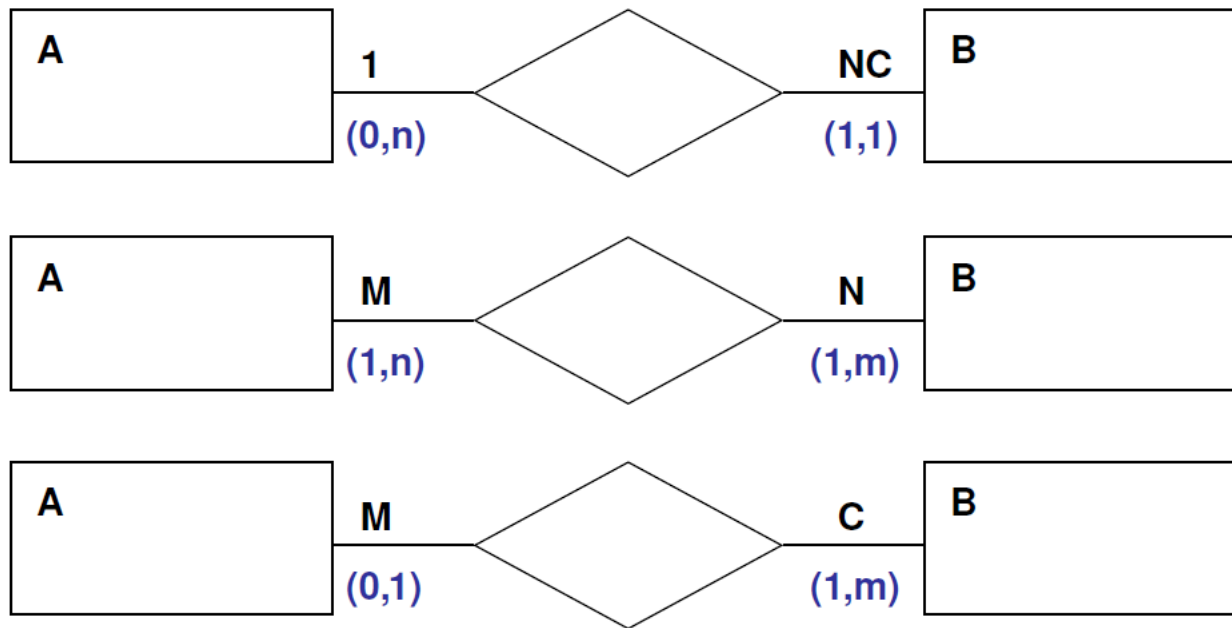
Notatia
numerica

Notatia
crow's foot
(Martin)

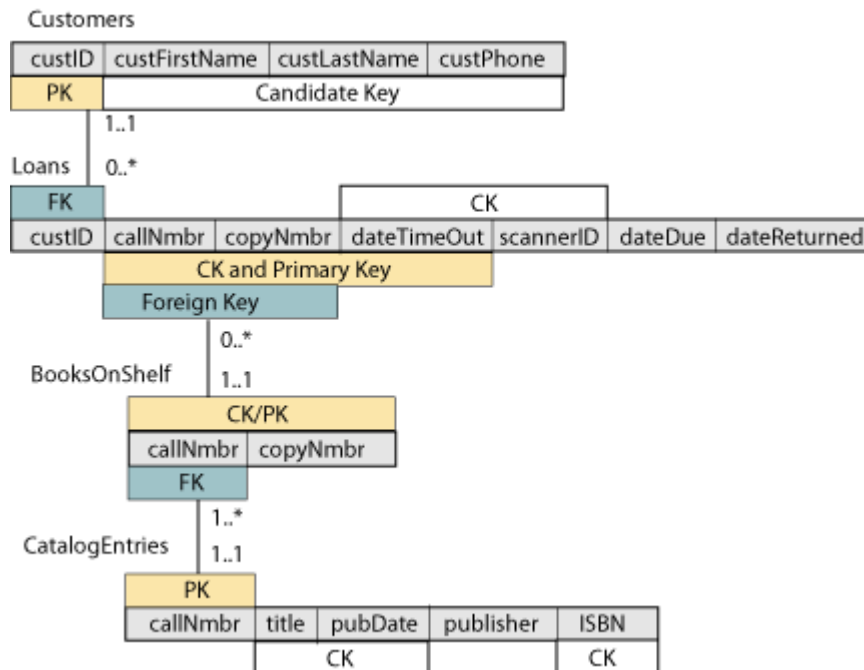
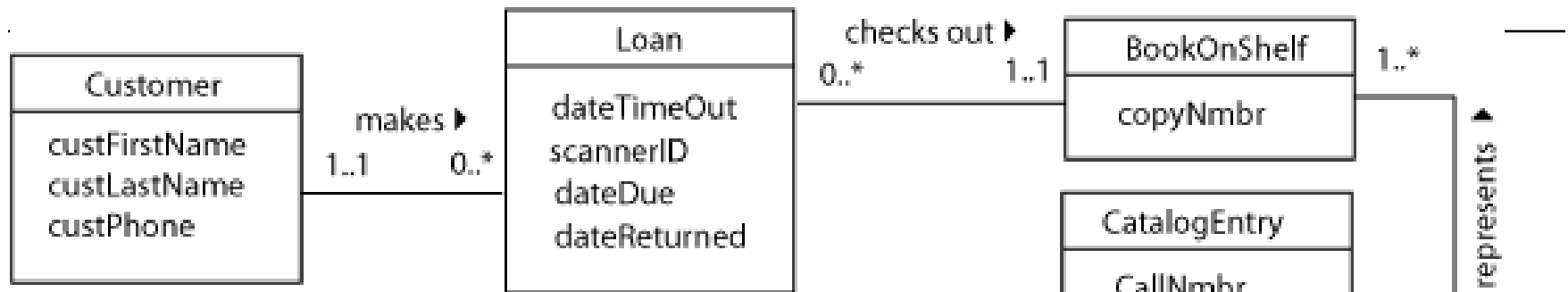
Notatia
sageti

Notatia
Bachmann

Reprezentare cardinalitate alternativa



UML Notation





Bibliografie

- <http://www.tomjewett.com/dbdesign/>