

# Modelul Entitate-Relație

Diagrame E/R

Entitate “Weak”

Obținerea Schemei Relaționale din Diagrame E/R

# Modelul E/R

- Scop: permite schițarea elementelor ce vor fi păstrate în BD (ajută la definirea schemei BD indiferent de modelul de date ales).
  - Include constrângeri asupra datelor, dar nu include operații asupra datelor.
- Ce se obține: desen numit *diagramă entitate-relație*.  
(mai exact relație de legătură)
- **Ulterior**: se convertește diagrama E/R într-o schemă de BD.

# Modelarea E/R

- Proiectarea BD face parte din cadrul mai general de proiectare a unui sistem informatic destinat unei companii.
- Există persoane din conducerea companiei care au cunoștințe despre afacerea respectivă, dar nu pot spune ce să conțină BD.
- Schițarea componentelor cheie reprezintă o modalitate eficientă pentru dezvoltarea unei BD funcționale.

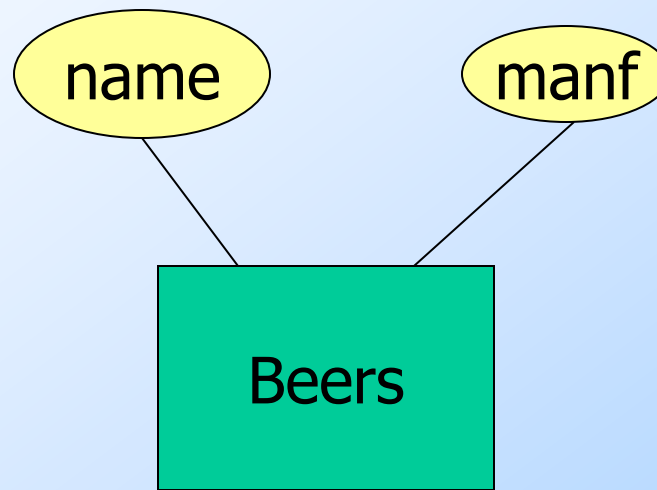
# Seturi de Entități

- *Entitate* = un “lucru” sau obiect din lumea reală.
- *Set de Entități* = colecție de entități similare.
  - Asemănător cu o clasă din programarea orientată obiect.
- *Atribut* = proprietate a unui set de entități.
  - Atributele sunt valori simple, ca de exemplu întregi sau șiruri de caractere;
  - NU sunt valori compuse ca de exemplu structuri, seturi, etc.

# Diagrame E/R

- Într-o diagramă entitate-relație :
  - Entitate = dreptunghi.
  - Atribut = oval, cu o linie către un dreptunghi ce reprezintă setul entitate corespunzător.

# Exemplu:

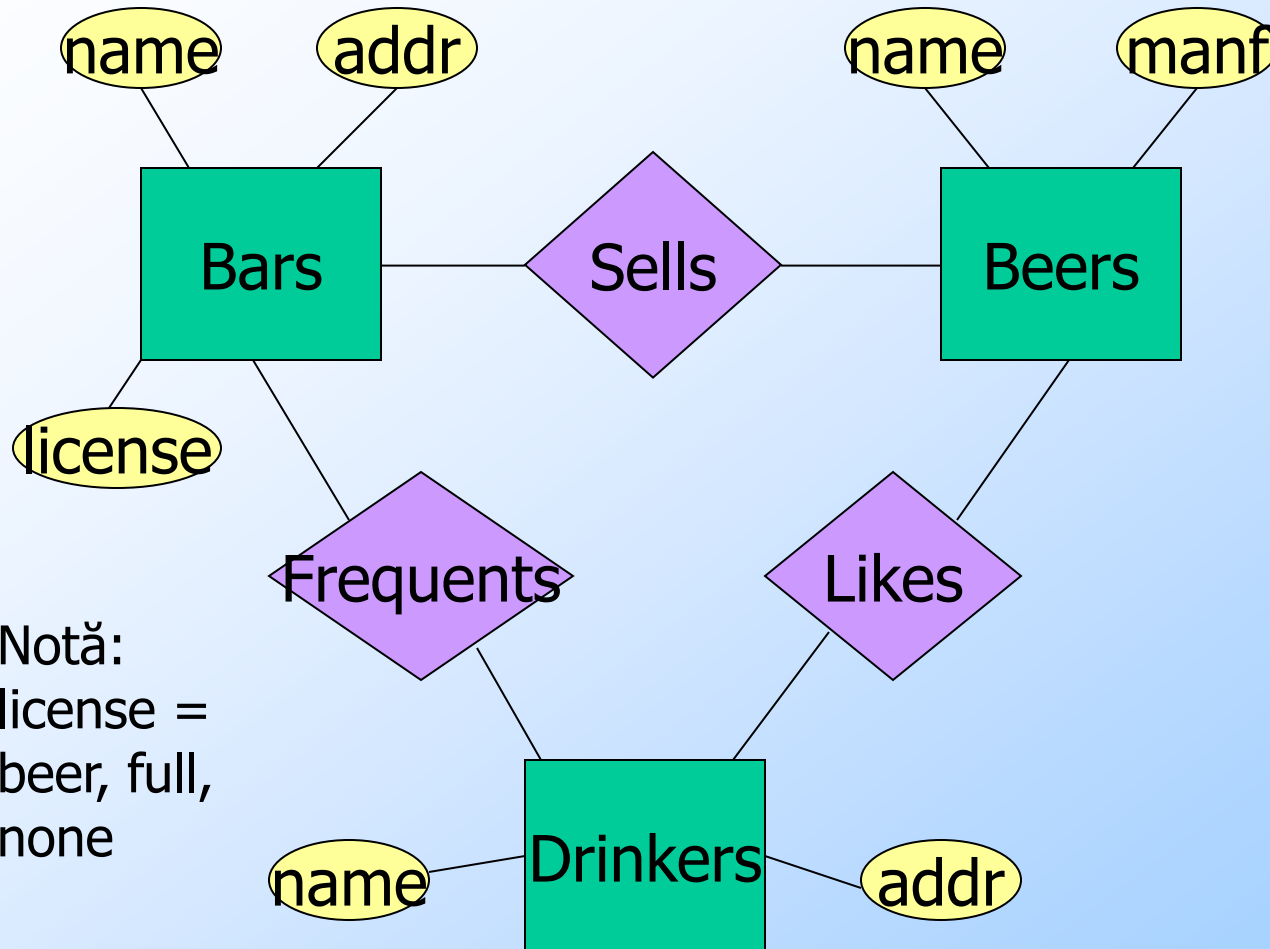


- Setul entitate **Beers** are două attribute, **name** și **manf** ("manufacturer").
- Fiecare entitate **Beers** are valori pentru aceste două attribute, de exemplu (Bud, Anheuser-Busch)

# Relații de legătură

- O relație de legătură conectează două sau mai multe seturi de entități.
- Este reprezentată printr-un romb, cu linii la fiecare din seturile de entități implicate.

# Exemplu: Relații de legătură



"Bars Sell some Beers".

"Drinkers Like some Beers".

"Drinkers Frequent some Bars".



# Setul Relație de legătură

- “Valoarea” curentă a unui *set entitate* reprezintă setul entităților aparținătoare.
  - **Exemplu:** setul tuturor barurilor din BD.
- “Valoarea” unei relații de legătură este un *set relație de legătură*, un set de tuple cu o componentă din fiecare set entitate ce intră în legătură.

# Exemplu: Set Relație de legătură

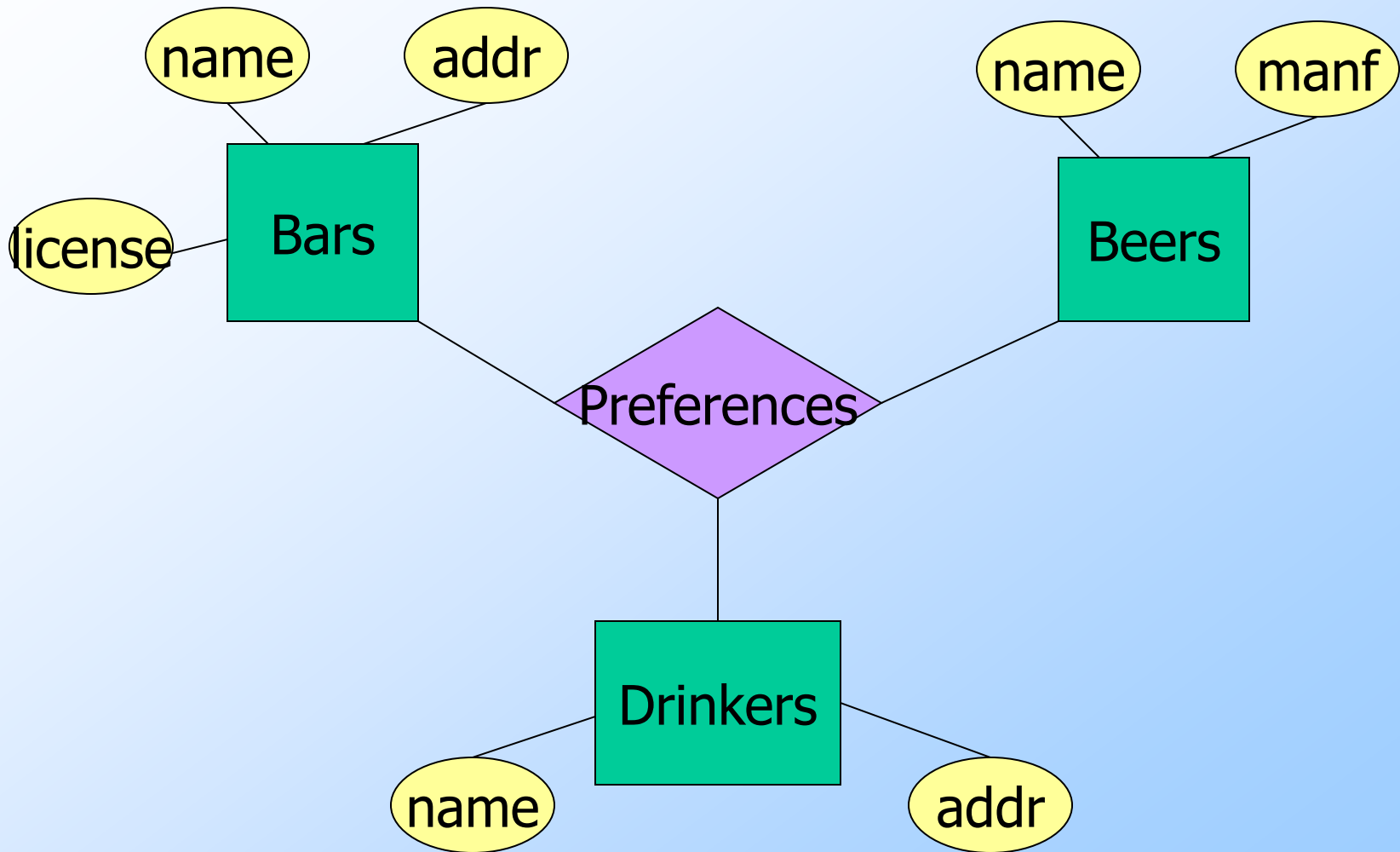
- Pentru relația de legătură **Sells**, se poate imagina setul relație de legătură următor:

Bar	Beer
Joe's Bar	Bud
Joe's Bar	Miller
Sue's Bar	Bud
Sue's Bar	Pete's Ale
Sue's Bar	Bud Lite

# Relații de legătură Multiple

- Uneori este nevoie ca o relație de legătură să conecteze mai mult de două seturi entitate.
- Să presupunem că persoanele (drinkers) vor consuma anumite beri (beers) la anumite baruri (bars).
  - Cele trei relații de legătură binare **Likes**, **Sells** și **Frequents** nu permit să se facă această distincție.
  - Este nevoie de o relație de legătură cu 3 capete.

# Exemplu: Relație de legătură Multiplă



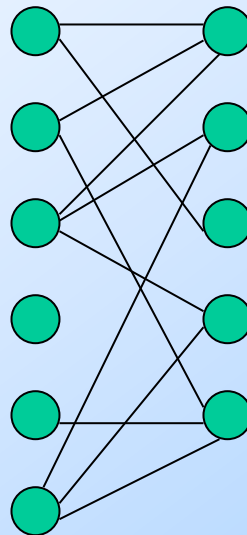
# Un Set Tipic Relație de Legătură Multiplă

Bar	Drinker	Beer
Joe's Bar	Ann	Miller
Sue's Bar	Ann	Bud
Sue's Bar	Ann	Pete's Ale
Joe's Bar	Bob	Bud
Joe's Bar	Bob	Miller
Joe's Bar	Cal	Miller
Sue's Bar	Cal	Bud Lite

# Relații de Legătură M:N

- Ținta: relații de legătură **binare**, cum este **Sells** între **Bars** și **Beers**.
- Într-o **relație de legătură M:N** (*many-many*), o entitate din oricare set poate fi conectată cu mai multe entități din celălalt set.
  - De exemplu, un bar vinde mai multe sortimente de bere; o marcă de bere este vândută în mai multe baruri.

# M:N

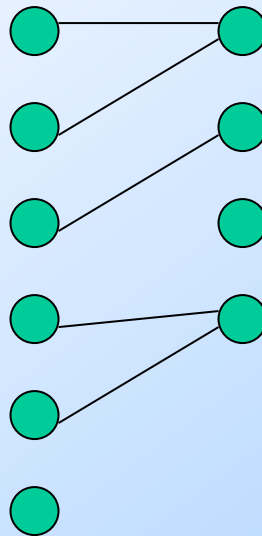


# Relații de Legătură N:1

- Unele relații de legătură binare sunt **N:1** (*many-one*).
- Fiecare entitate din primul set se conectează cu cel mult o entitate din setul al doilea.
- O entitate din setul al doilea se poate conecta cu zero, una, sau mai multe entități din primul set.



# N:1



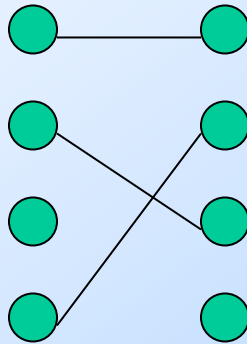
# Exemplu: Relație de Legătură N:1

- Favorite, de la Drinkers la Beers este “many-one”.
- O persoană are cel mult 1 marcă de bere favorită.
- Dar o marcă de bere poate fi berea favorită a mai multor persoane, inclusiv zero.

# Relații de Legătură 1:1

- Într-o **relație de legătură 1:1** (*one-one*), fiecare entitate a unuia din seturile entitate este pus în relație cu cel mult o entitate a celuilalt set.
- **Exemplu:** Relația de legătură **Best-seller** între seturile entitate **Manfs** ("manufacturer") și **Beers**.
  - O marcă de bere nu poate fi fabricată de mai mult de un fabricant, și nici un fabricant nu poate avea mai mult de o marcă de bere cel mai bine vândută (presupunând că nu sunt două sau mai multe la egalitate pe primul loc).

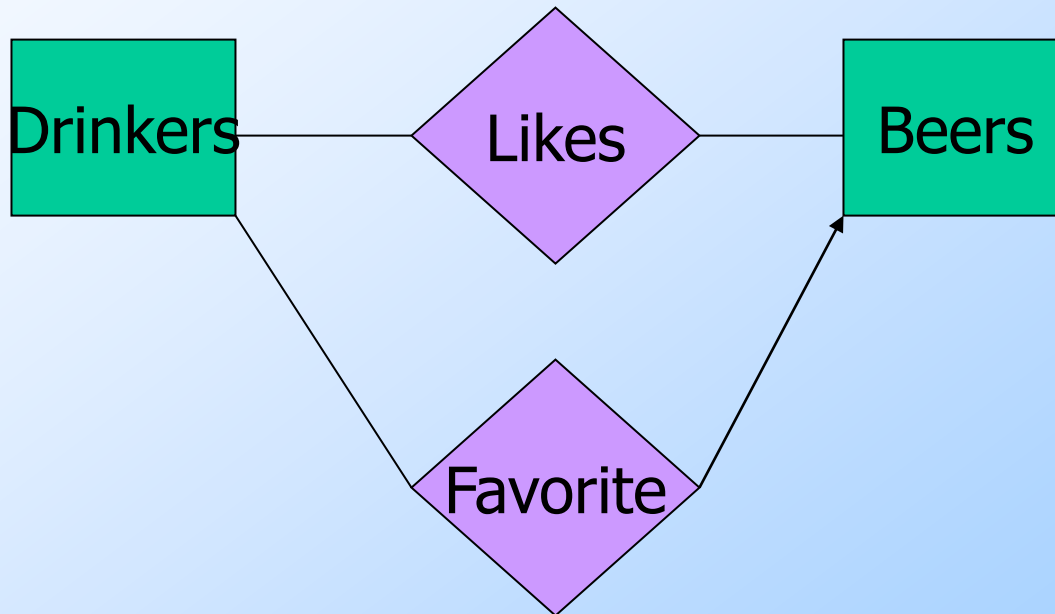
1:1



# Reprezentarea capătului relației de legătură

- Se folosește săgeata cu vârf spre capătul "1" al relației de legătură "many-one".
  - De ținut minte: Seamănă cu dependența funcțională.
- O relație de legătură 1:1 se reprezintă cu săgeți la ambele capete (spre ambele seturi entitate).
- Săgeată rotunjită = "exact 1," de exemplu, când fiecare entitate din primul set este pusă în relație cu exact 1 entitate din setul al doilea.

# Exemplu: Relație de Legătură N:1



Notă: două relații de legătură conectează aceleași seturi entitate, dar sunt diferite.

# Exemplu: Relație de Legătură 1:1

- Se consideră **Best-seller** între **Manfs** și **Beers**.
- Anumite mărci de bere nu sunt cele mai bine vândute mărci de bere ale nici unui fabricant, prin urmare vârful săgeții spre **Manfs** nu poate fi rotunjit.
- Dar un fabricant de bere obligatoriu are 1 marcă de bere cel mai bine vândută.

# Diagrama E/R



O bere este "best-seller"  
pentru 0 sau 1  
"manufacturer".

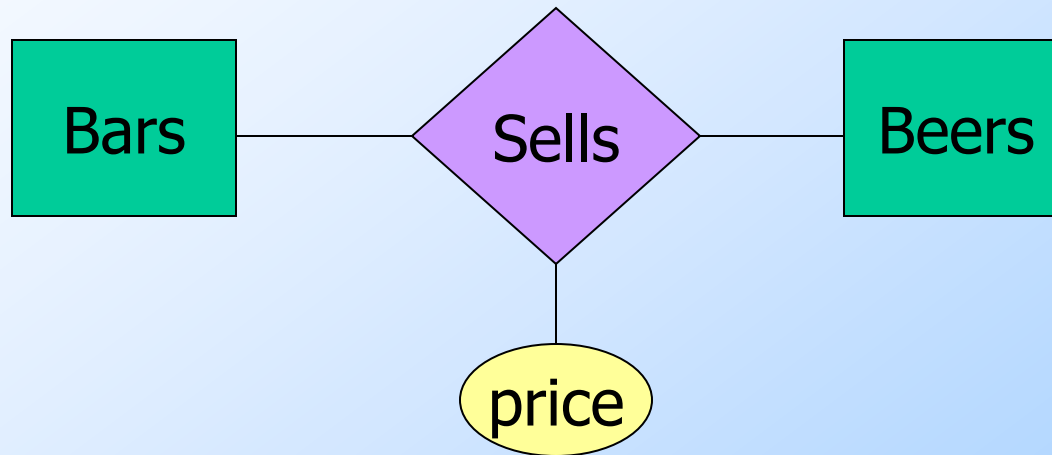
Un "manufacturer" are  
exact 1 "best-seller".



# Atribute pentru Relații de Legătură

- Uneori este folositor să se atașeze un atribut unei relații de legătură.
- Trebuie gândit la acest atribut ca la o proprietate a tuplelor din setul relație de legătură.

# Exemplu: Atribut pentru o Relație de Legătură

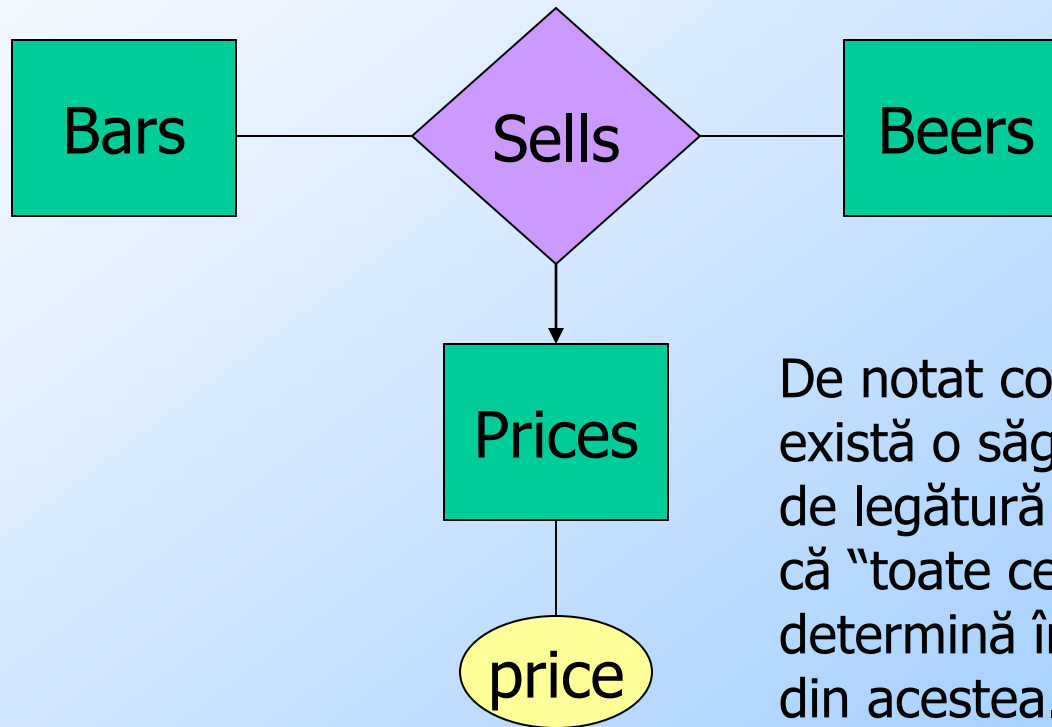


Preț este o funcție ce depinde atât de bar cât și de bere, nu de una din ele.

# Diagrame Echivalente fără Attribute pentru Relațiile de Legătură

- Se crează un set entitate pentru a reprezenta valorile atributului.
- Acel set entitate participă în relația de legătură.

# Exemplu: Eliminarea unui Atribut de la o Relație de Legătură



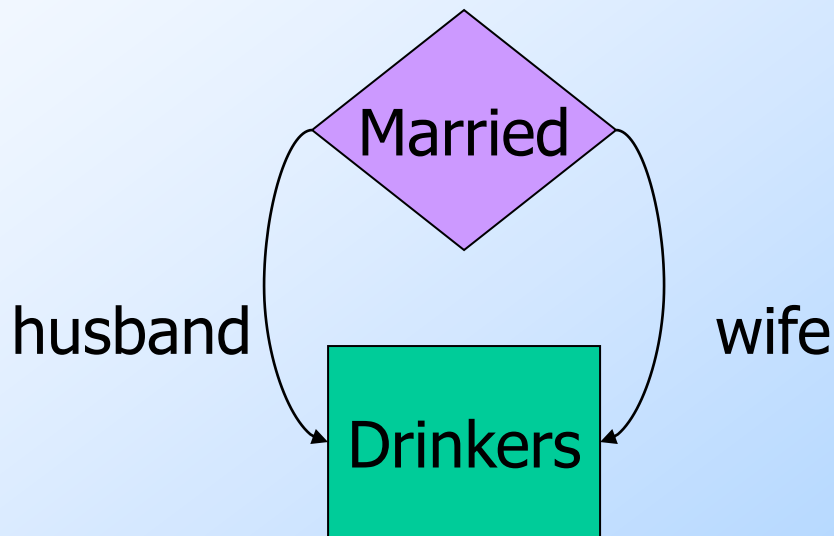
De notat convenția: dacă există o săgeată de la relația de legătură multiplă, înseamnă că "toate celelalte seturi entitate determină împreună unul singur din acestea."

# Roluri

- Uneori un set entitate apare de mai multe ori într-o relație de legătură.
- Se etichetează capătele relației de legătură către setul entitate cu nume ce semnifică *roluri*.

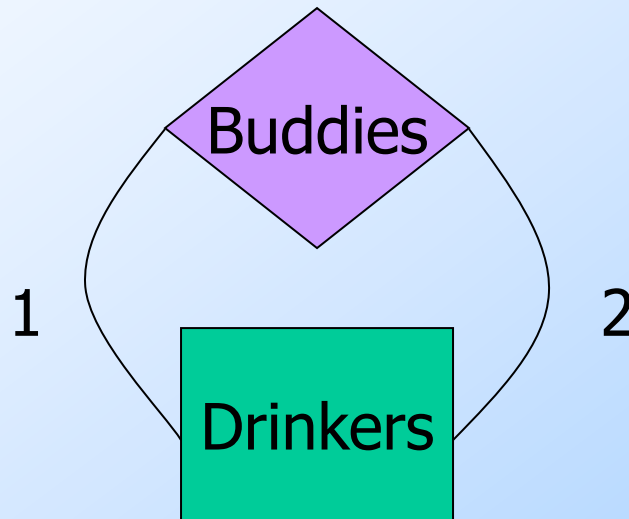
# Exemplu: Roluri

Setul Relație de Legătură



Husband	Wife
Bob	Ann
Joe	Sue
...	...

# Exemplu: Roluri



Setul Relație de Legătură

Buddy1	Buddy2
Bob	Ann
Joe	Sue
Ann	Bob
Joe	Moe
...	...

# Subclase

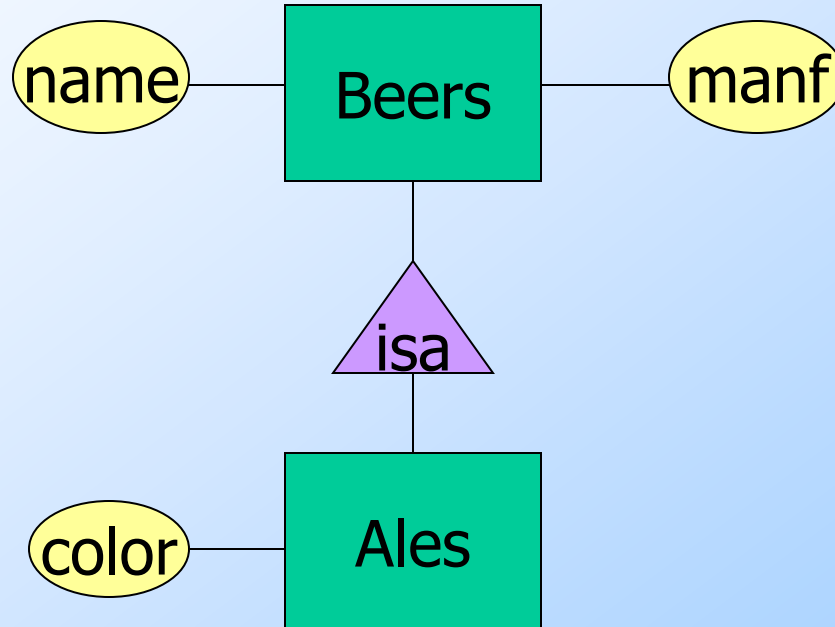
- *Subclasă* = caz special = mai puține entități = mai multe proprietăți.
- *Exemplu*: "Ale" reprezintă un sortiment de bere.
  - Nu fiecare bere este un "ale", dar unele sunt.
  - Să presupunem că în plus față de toate *proprietățile* (attribute și relații de legătură) lui "beers", "ales" au atributul "color".



# Subclase în Diagrame E/R

- Să presupunem că subclasele formează un arbore.
  - Adică nu există moștenire multiplă.
- Triunghiuri “Isa” indică relația de legătură subclasă.
  - Pointează spre superclasă.

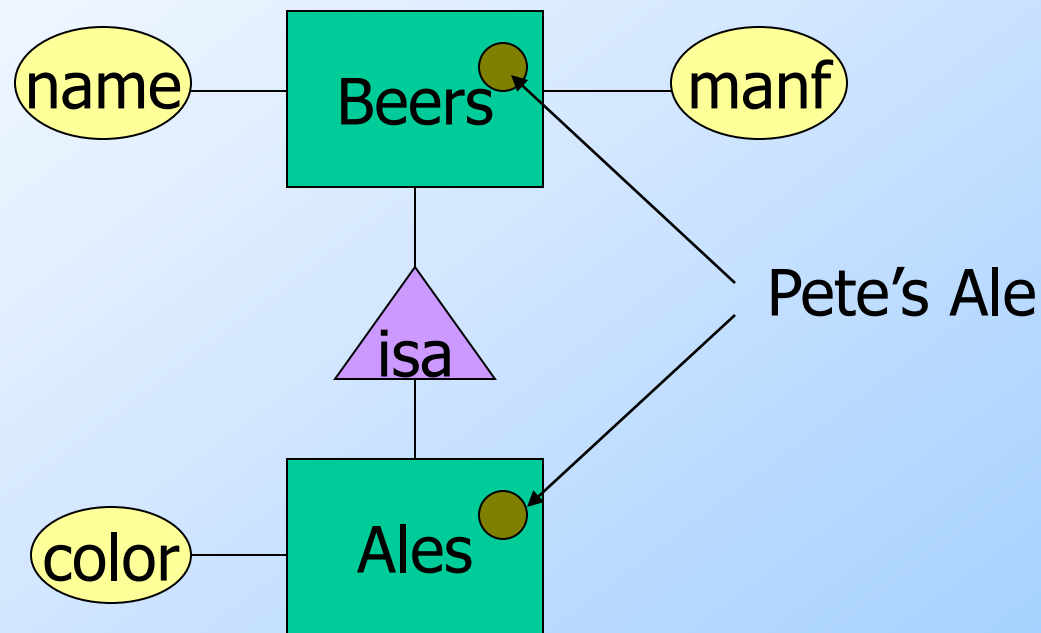
# Exemplu: Subclase



# E/R Vs. Subclase Orientate-Obiect

- În OO, obiectele sunt într-o clasă și numai una.
  - Subclasele moștenesc de la superclase.
- În contrast, entitățile E/R au *reprezentante* în toate subclasele la care aparțin.
  - **Regulă**: dacă entitatea  $E$  este reprezentată într-o subclasă, atunci  $E$  este reprezentată în superclasă (și recursiv în sus în arbore).

# Exemplu: Reprezentante ale Entităților



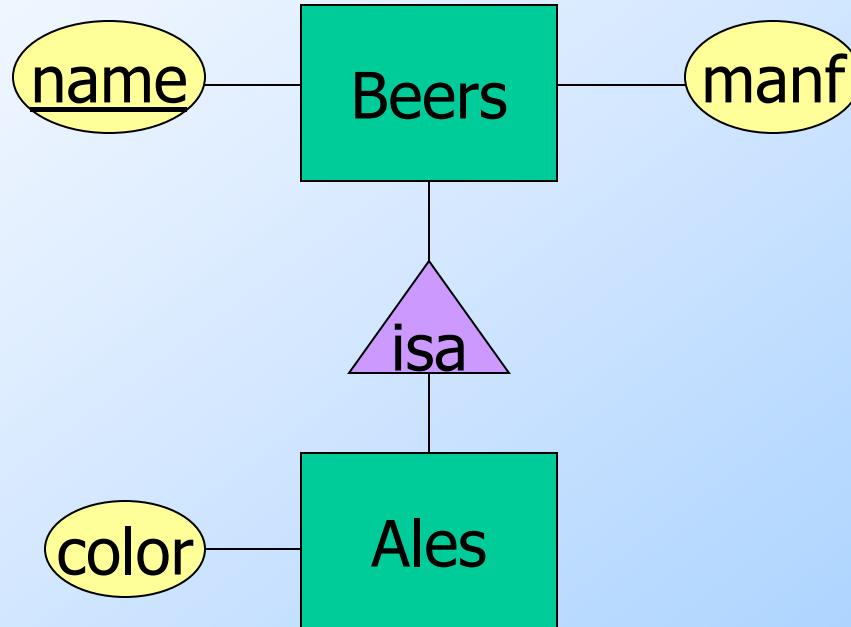
# Chei

- O *cheie* reprezintă o mulțime de attribute pentru un set entitate astfel încât două entități din set nu pot avea aceleași valori pentru toate attributele cheii.
- Se poate ca două entități să aibă anumite valori egale, pentru attributele cheii, dar nu toate.
- Trebuie desemnată o cheie pentru fiecare set entitate.

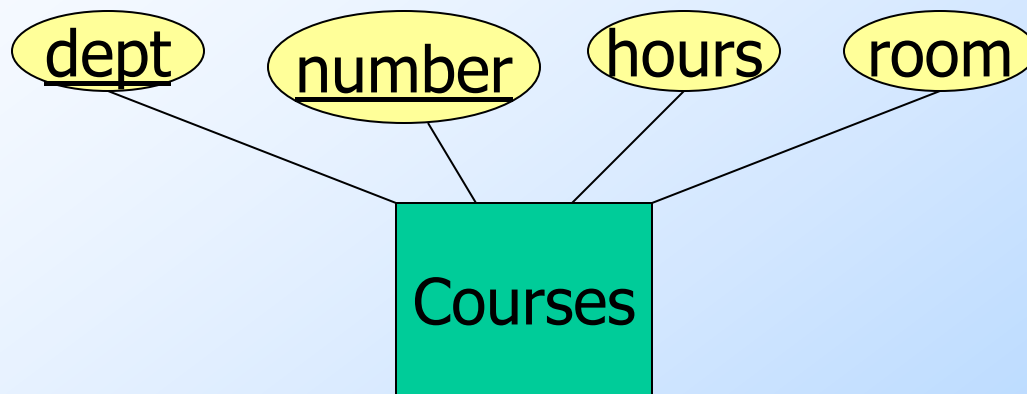
# Chei în Diagrame E/R

- Se subliniază atributul(-ele) cheie.
- Într-o ierarhie "Isa", doar setul entitate rădăcină are cheie, și aceasta servește ca și cheie pentru toate entitățile din ierarhie.

Exemplu: **name** este Cheie  
pentru **Beers**



# Exemplu: Cheie Multi-atribut



- De notat că **hours** și **room** pot fi alese pentru cheie, dar este nevoie de o singură cheie.



# Seturi Entitate "Weak"

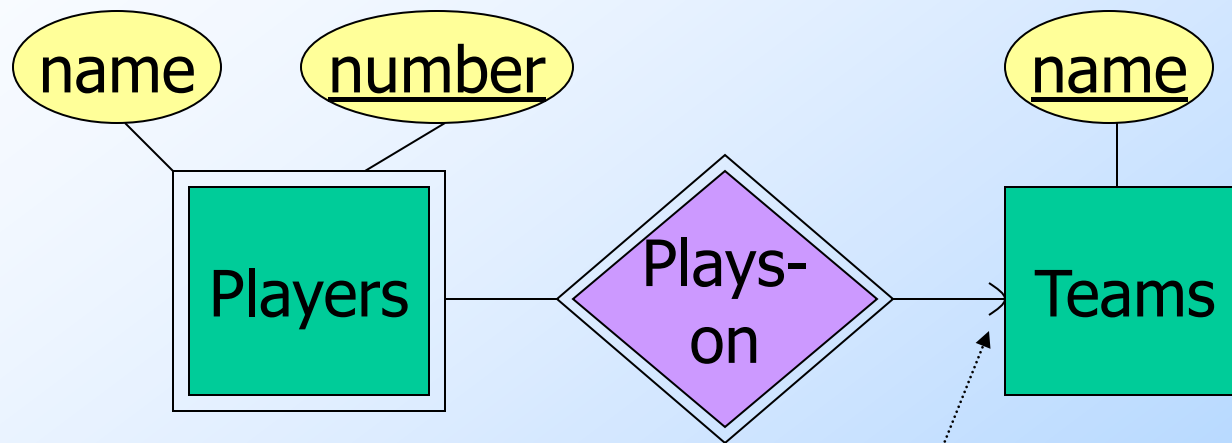
- Ocazional, entitățile unui set entitate necesită "ajutor" pentru a le identifica în mod unic.
- Setul entitate  $E$  se spune că este "*weak*" dacă pentru a identifica unic entitățile din  $E$ , este nevoie de a urmări relații de legătură "many-one" plecând de la  $E$  și să se includă cheia entităților din seturile entitate conectate.

# Exemplu: Set Entitate "Weak"

- "name" este aproape o cheie pentru jucători de fotbal, dar pot exista doi jucători cu același "name".
- "number" este un lucru ce nu poate fi cheie, deoarece jucători din două echipe pot avea același "number" (numărul de pe tricou).
- Dar numărul de pe tricou, împreună cu numele echipei pus în legătură de "Plays-on" ar trebui să fie unic.

# Diagrama E/R

## Set Entitate "Weak"



Notă: trebuie rotunjit deoarece fiecare jucător are nevoie de o echipă pentru cheie.

Linie dublă pentru rombul relației de legătură N:1 *sustinătoare*.  
Linie dublă pentru dreptunghiul setului entitate "weak".

# Reguli pentru Setul Entitate “Weak”

- Un set entitate “weak” are una sau mai multe relații de legătură N:1 către alte seturi entitate (susținătoare).
- Nu fiecare relație de legătură N:1 ce pleacă de la un set entitate “weak” este “susținător”.
- Relațiile de legătură susținătoare trebuie să aibă vârful de săgeată rotunjit (entitatea de la capătul “one” este garantată).

# Reguli pentru Setul Entitate “Weak”

- Cheia pentru un set entitate weak este format din attributele subliniate și din cheile seturilor entitate susținătoare.
- De exemplu **number** (player) și **name** (team) este o cheie pentru **Players** în exemplul precedent.

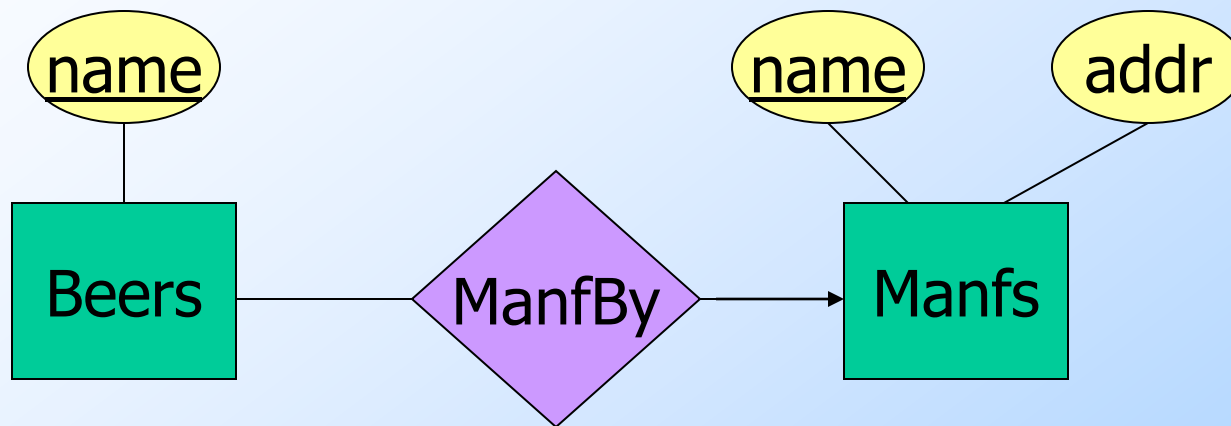
# Tehnici de Proiectare

1. A se evita redundanța.
2. A se limita utilizarea de seturi entitate "weak".
3. Să nu se folosească un set entitate dacă se poate folosi un atribut.

# Evitarea Redundanței

- *Redundanță* = specificarea aceluiași lucru în două (sau mai multe) moduri diferite.
- Se irosește spațiu și (mai important) se încurajează inconsistența.
  - Două reprezentări ale aceluiași fapt devin inconsistente dacă se modifică una și se lasă cealaltă nemodificată.
  - De reamintit anomaliiile de la DF-le.

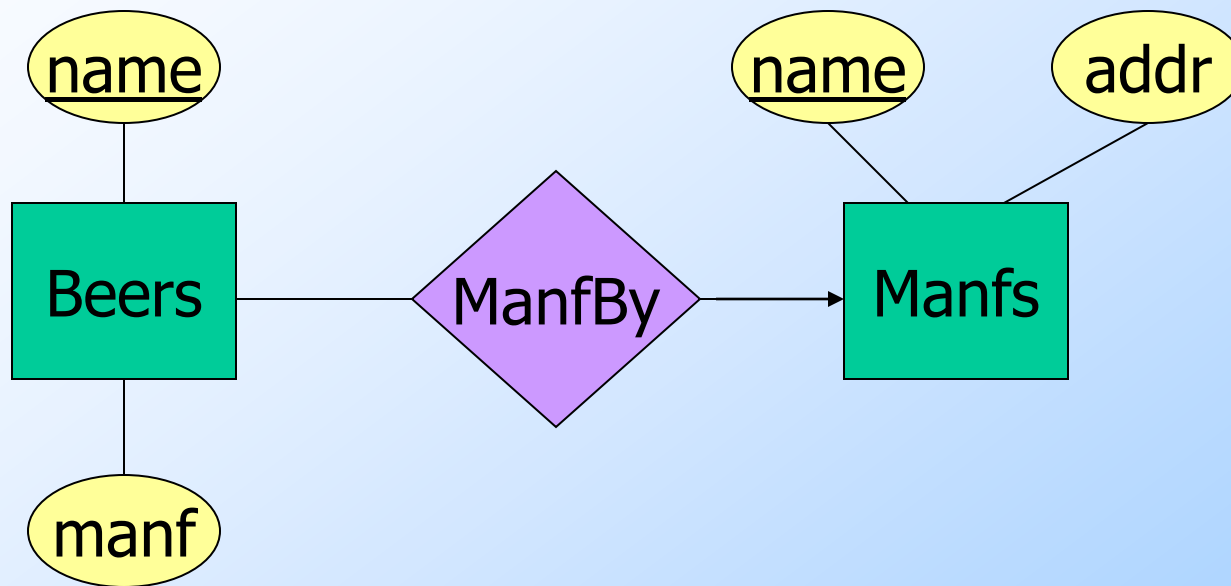
# Exemplu: Bun



Această proiectare specifică adresa fiecărui fabricant exact o singură dată.

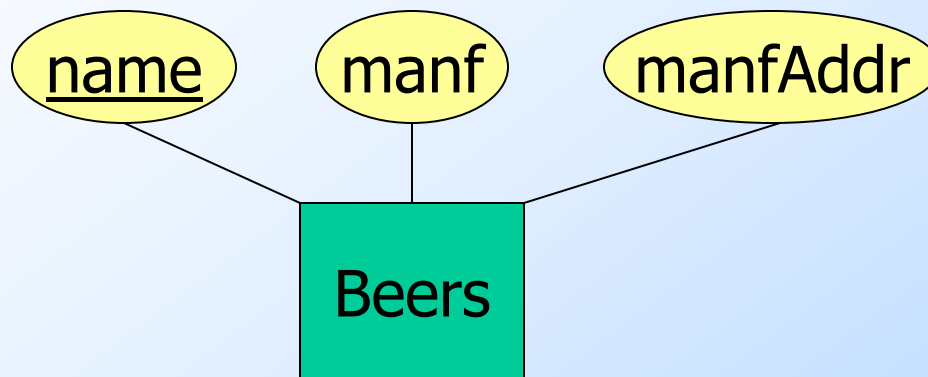


# Exemplu: Rău



Această proiectare precizează fabricantul unei mărci de bere de două ori: prima dată ca un atribut și a doua oară ca o entitate pusă în legătură.

# Exemplu: Rău



Această proiectare repetă adresa fabricantului pentru fiecare bere și adresa se pierde dacă temporar nu există mărci de bere pentru un fabricant.

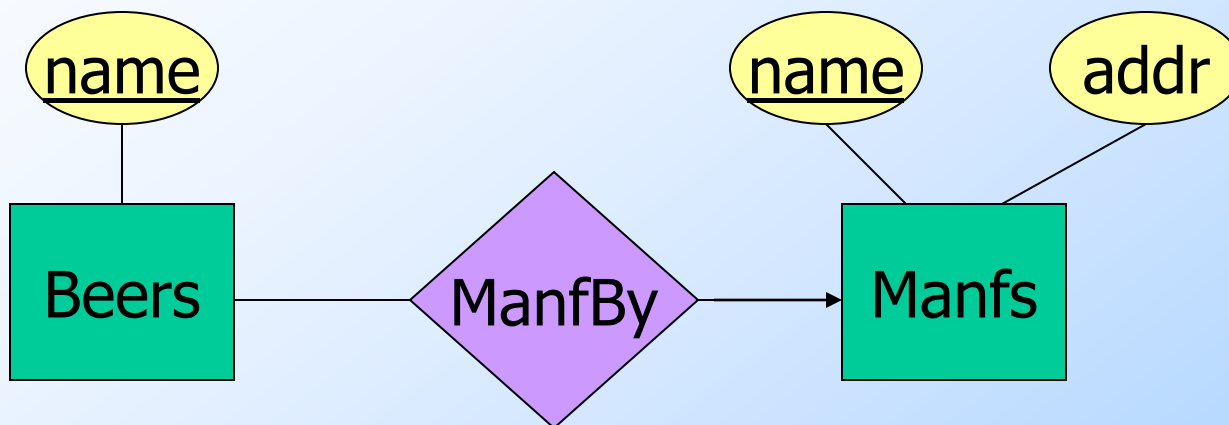
# Seturi Entitate Versus Atribute

- Un set entitate ar trebui să satisfacă cel puțin una din următoarele condiții:
  - Reprezintă mai mult decât numele unui lucru; are cel puțin un atribut ce nu este cheie.

sau

- Este capătul "many" într-o relație de legătură "many-one" sau "many-many".

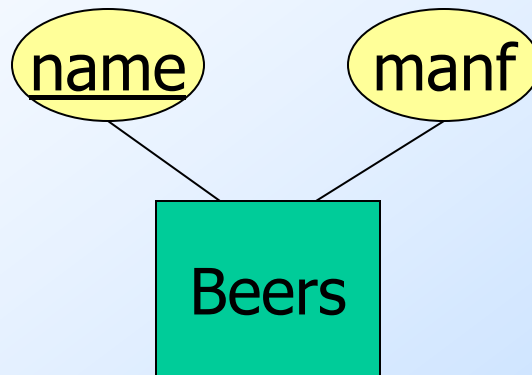
# Exemplu: Bun



**Manfs** merită să fie un set entitate deoarece are atributul ce nu este cheie **addr**.

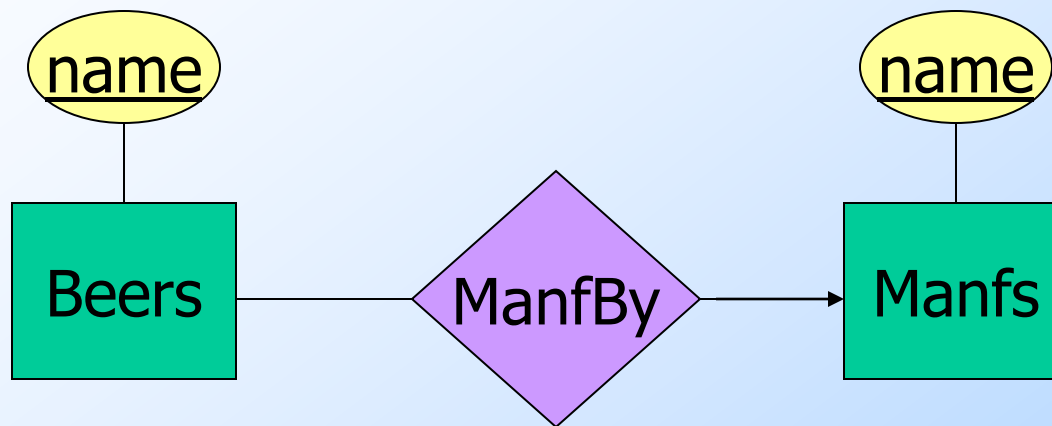
**Beers** merită să fie un set entitate deoarece este la capătul "many" al relației de legătură "many-one" **ManfBy**.

# Exemplu: Bun



Nu este nevoie să se definească "manufacturer" ca un set entitate, deoarece nu se înregistrează nimic despre fabricanți în afara numelui lor.

# Exemplu: Rău



Deoarece "manufacturer" nu este altceva decât un nume, și nu este la capătul "many" al nici unei relații de legătură, nu ar trebui să fie set entitate.

# Să nu se exagereze cu folosirea Seturilor Entitate "Weak"

- Proiectanții de BD începători au adesea îndoieli în a alege cheia.
  - Aceștia marchează seturile entitate ca fiind "weak", susținute de toate seturile entitate înconjurătoare cu care sunt legate.
- În realitate, se crează ID-uri unice pentru seturi entitate.
  - Exemple: CNP pentru o persoană, număr serial pentru un calculator etc.

# Când este nevoie de Seturi Entitate “Weak”?

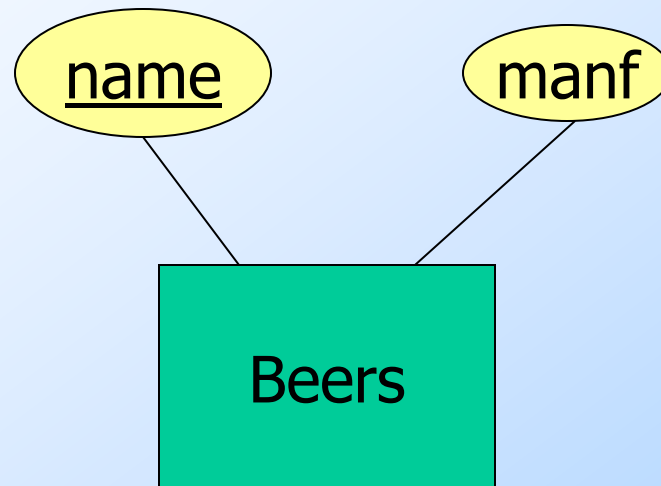
- Motivul obișnuit este acela că nu există o autoritate globală capabilă să creeze ID-uri unice.
- **Exemplu:** este puțin probabil ca să existe o înțelegere între echipele de fotbal din întreaga lume pentru a atribui un număr unic fiecărui jucător.



# Obținerea Schemei Relaționale din Diagrame E/R

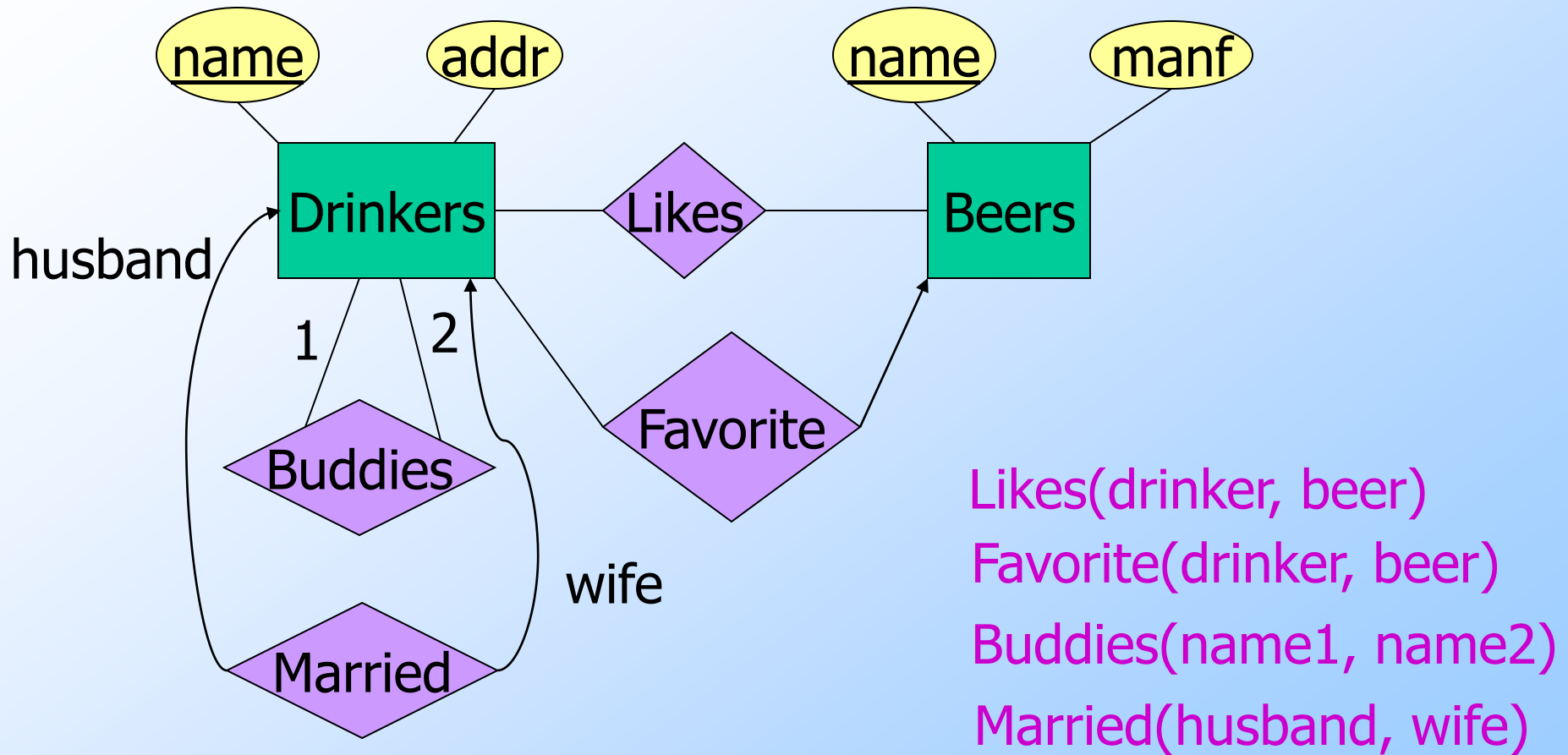
- Set entitate -> relație.
  - Attribute -> atribut.
- Relații de legătură -> relații cu următoarele atribut:
  - Cheile seturilor entitate conectate.
  - Atributele relației de legătură (dacă există).

# Set Entitate -> Relație



Relația: Beers(name, manf)

# Relație de Legătură -> Relație



# Combinarea Relațiilor

- Se poate combina într-o relație:
  1. Relația pentru un set entitate  $E$
  2. Relațiile pentru relațiile de legătură "many-one" pentru care  $E$  este capătul "many."
- **Exemplu:** Drinkers(name, addr) și Favorite(drinker, beer) se combină pentru a forma Drinker1(name, addr, favBeer).

# Risc implicat de Relații de Legătură “Many-Many”

- Combinarea Drinkers cu Likes este greșit. Conduce la redundanță:

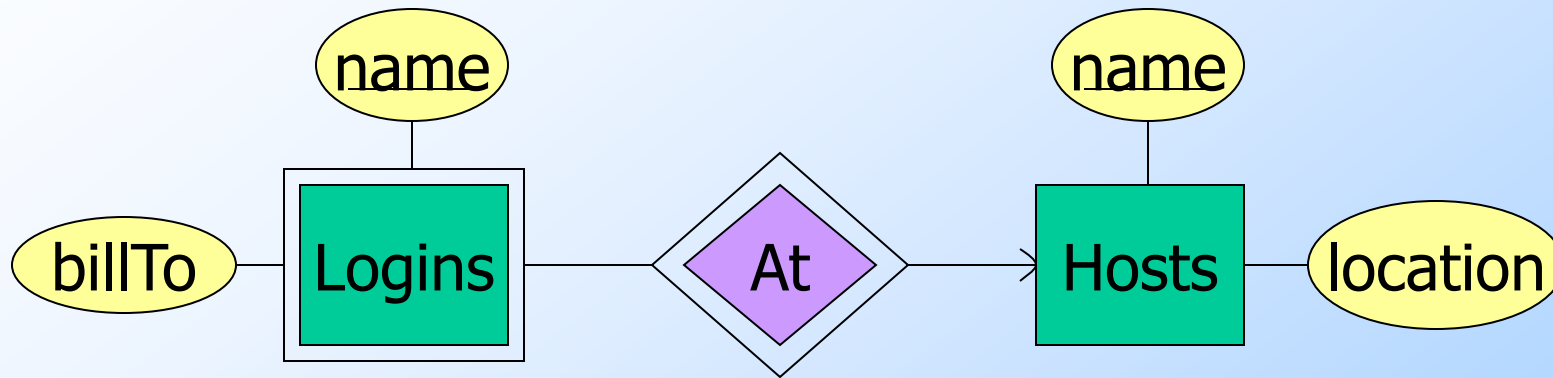
name	addr	beer
Sally	123 Maple	Bud
Sally	123 Maple	Miller

Redundanță

# Seturi Entitate "Weak"

- Relația pentru un set entitate "weak" trebuie să includă attributele ce compun cheia (inclusiv cele ce aparțin altor seturi entitate), plus propriile attribute noncheie.
- O relație de legătură susținătoare este redundantă și nu conduce la nici o relație (cu excepție când are attribute proprii).

# Exemplu: Set Entitate "Weak" -> Relație



Hosts(hostName, location)

Logins(loginName, hostName, billTo)

~~At(loginName, hostName, hostName2)~~

Trebuie să fie același

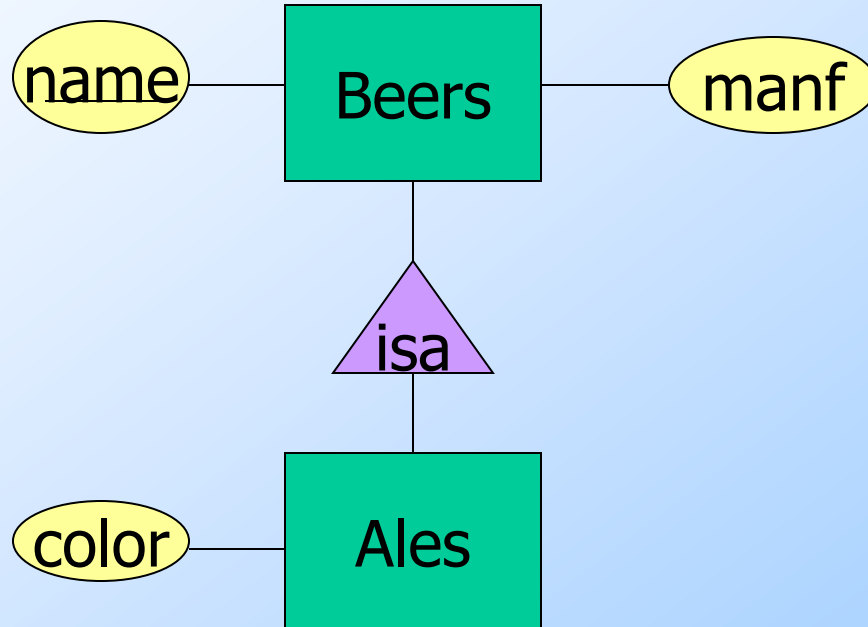
At devine parte  
a Logins

# Subclase: Trei Abordări

1. *Orientat-Obiect* : O relație per subset de subclase, cu toate atributele relevante.
2. *Folosire null-uri* : O relație; entitățile au NULL în atributele ce nu le aparțin.
3. *Stil E/R* : O relație pentru fiecare subclasă:
  - Atribut (-e) cheie.
  - Atribute ale acelei subclase.



# Exemplu: Subclasă -> Relații



# Orientat-Obiect

name	manf
Bud	Anheuser-Busch

Beers

name	manf	color
Summerbrew	Pete's	dark

Ales

Prinde bine la interogări de genul:

Găsește culoarea "ales" produsă de "Pete's".

# Stil E/R

name	manf
Bud	Anheuser-Busch
Summerbrew	Pete's

Beers

name	color
Summerbrew	dark

Ales

Prinde bine la interogări de genul:

Găsește toate "beers" (inclusiv "ales") produse de "Pete's".

# Folosire Null-uri

name	manf	color
Bud Summerbrew	Anheuser-Busch Pete's	NULL dark

Beers

Salvează spațiu dacă NU sunt multe  
atribute ce conțin NULL.