

# Adatbázisok 1.

## Egyed-kapcsolat modell – 4. rész

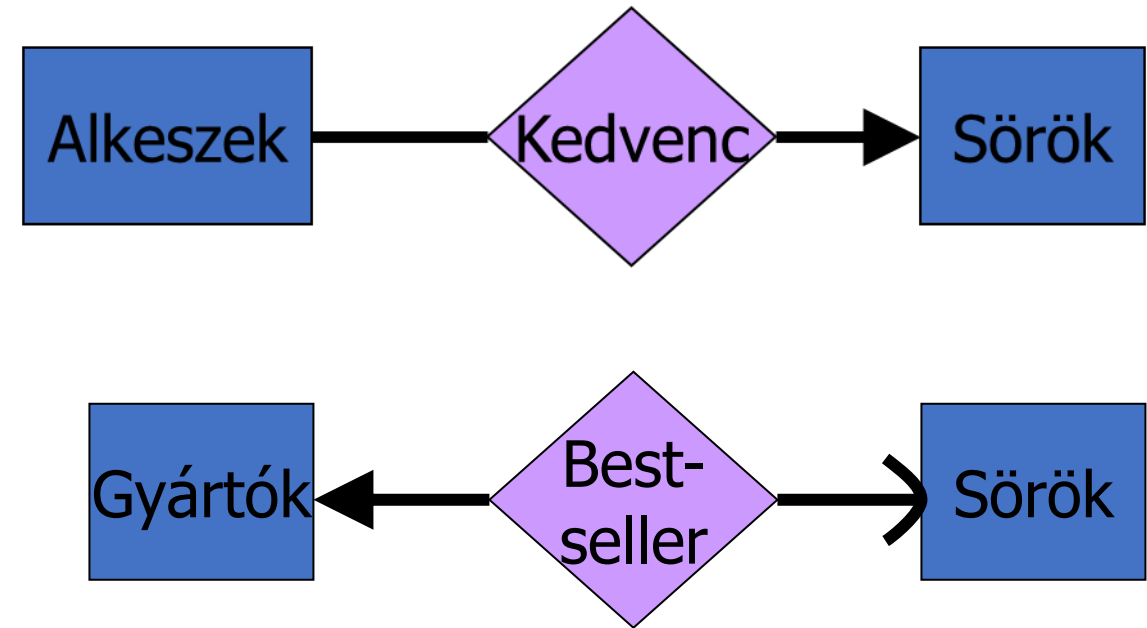
E/K diagramok

Gyenge egyedhalmazok

E/K diagramok átírása relációsémákká

# Ismétlés

- Egyed-kapcsolat diagram
- Kapcsolatok „számossága”
- Szerepek
- Alosztályok
- Gyenge egyedhalmazok



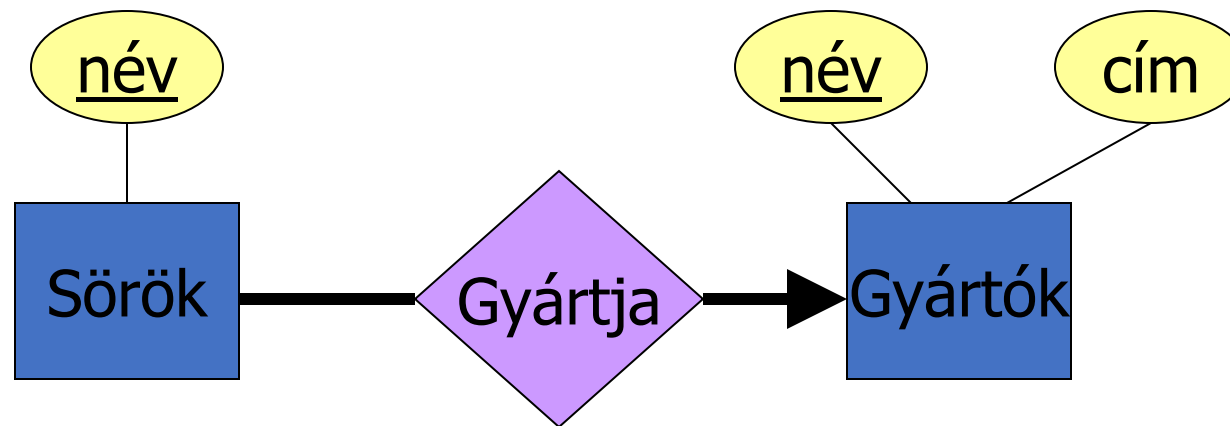
# Tervezési technikák

1. Redundancia elkerülése.
2. A gyenge egyedhalmazok óvatos használata.
3. Ne használjunk egyedhalmazt, ha egy attribútum éppúgy megfelelne a célnak.

# Redundancia elkerülése

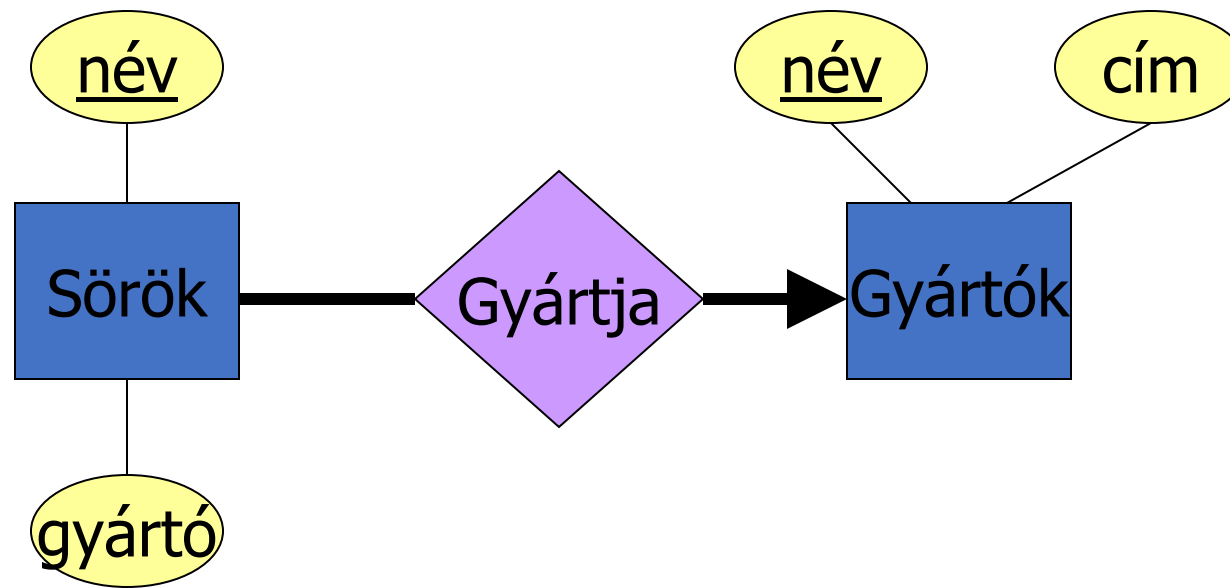
- *Redundancia* = ugyanazt a dolgot többféle módon is feltüntetjük.
- Ez egyrészt helypazarlás, másrészt növeli az inkonzisztencia veszélyét.
  - Ugyanazon tény két leírása akkor válik inkonzisztenssé, ha az egyik értéket megváltoztatjuk, míg a másiktól megfelelkezünk.
  - Emlékezzünk vissza a módosítási anomáliára.

## Példa: helyes



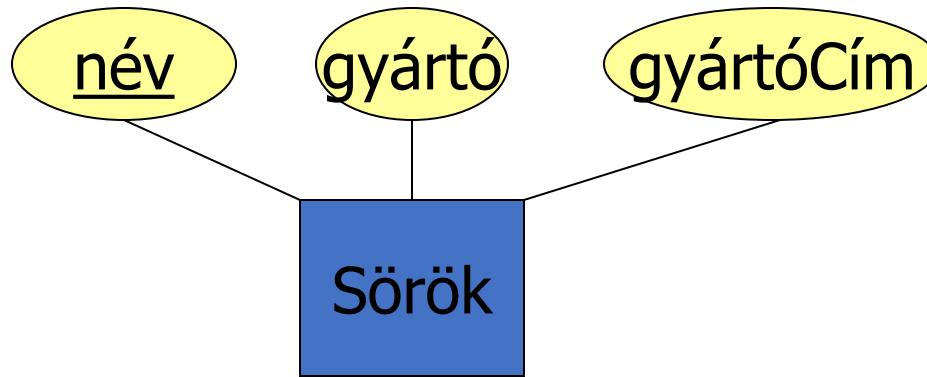
Ennél a diagramnál minden gyártónál pontosan egyszer szerepel a címe.

## Példa: helytelen



Ennél a diagramnál viszont egy-egy sör gyártója kétszer is szerepel: attribútumként és a kapcsolódó Gyártók egyedhalmazon keresztül.

## Példa: helytelen



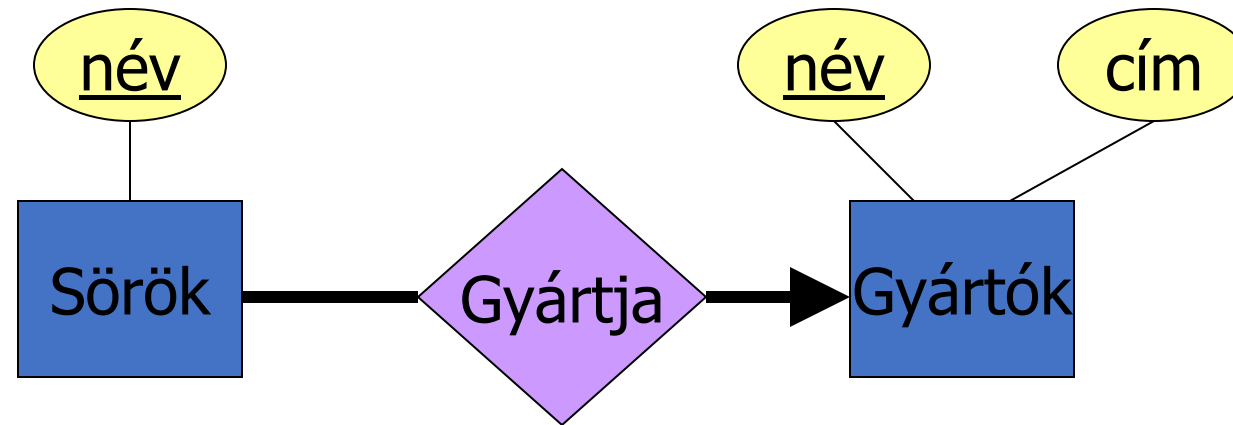
Ez a diagram minden sör esetén felsorolja a gyártó címét. Ha pedig ideiglenesen nincs megadva az adott gyártóhoz sör, akkor elveszítjük ezt a címet.

# Egyedhalmazok vs. attribútumok

- Egy egyedhalmaznak legalább egy feltételnek eleget kell tennie az alábbiak közül:
  - Többnek kell lennie, mint egy egyszerű név, azaz legalább egy nem kulcs attribútumának lennie kell.
- Vagy..
- a „sok” végén szerepel egy sok-egy kapcsolatnak.

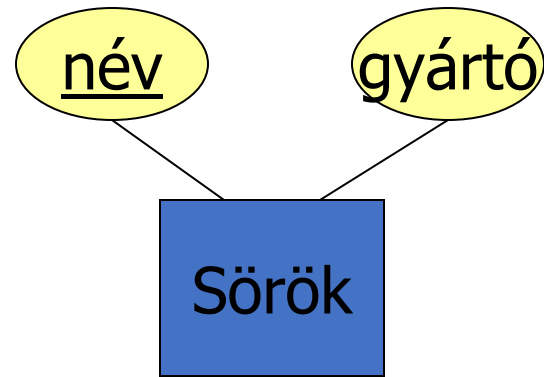


## Példa: helyes



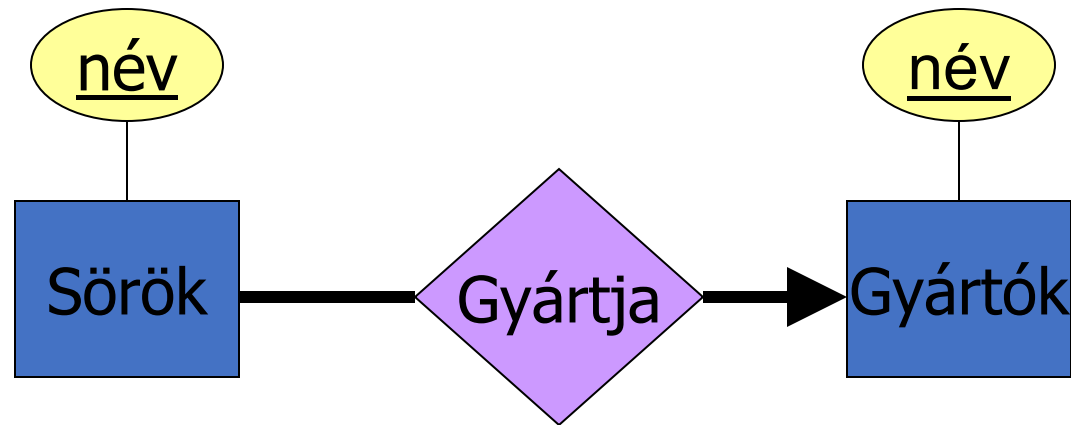
- A **Gyártóknak** valóban egyedhalmaznak kell lennie a nem kulcs **cím** attribútum miatt.
- A **Sörökre** ugyanez igaz, hiszen a **Gyártja** kapcsolat „sok” végén szerepel.

## Példa: helyes



Itt nem szükséges külön egyedhalmazt definiálni a gyártók számára, hiszen a nevükön kívül mást nem tárolunk.

## Példa: rossz



Mivel a gyártóknál csak a nevet tároljuk, ezen kívül a sok-egy kapcsolat „egy” oldalán szerepel, ezért fölösleges külön egyedhalmazt definiálni a számára.

# Ne használjunk gyenge egyedhalmazokat fölöslegesen

- Kezdő adatbázis tervezőknek gyakran kétségeik vannak abban tekintetben, hogy mit is lehet kulcsként definiálni, ami megállna önmaga jogán kulcsként.
  - Minden egyedhalmazt gyengének definiálnak, amit az összes vele összekötött egyedhalmaz támogat.
- A „nagybetűs életben” gyakran külön azonosítókat készítenek az egyedhalmazokhoz.
  - Pl. TAJ, rendszám tábla stb.

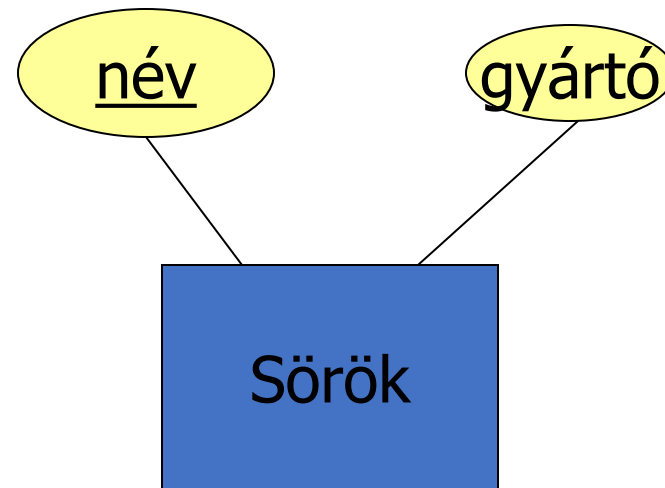
# Mikor van szükség gyenge egyedhalmazokra?

- A legtöbb esetben nem létezik olyan globális fórum, amely kioszthatná az egyedi azonosítókat.
- **Példa:** nem valószínű, hogy el lehetne érni, hogy a világ összes focijátékosát külön mezszámmal regisztrálják.

# E/K diagramok átírása relációsémává

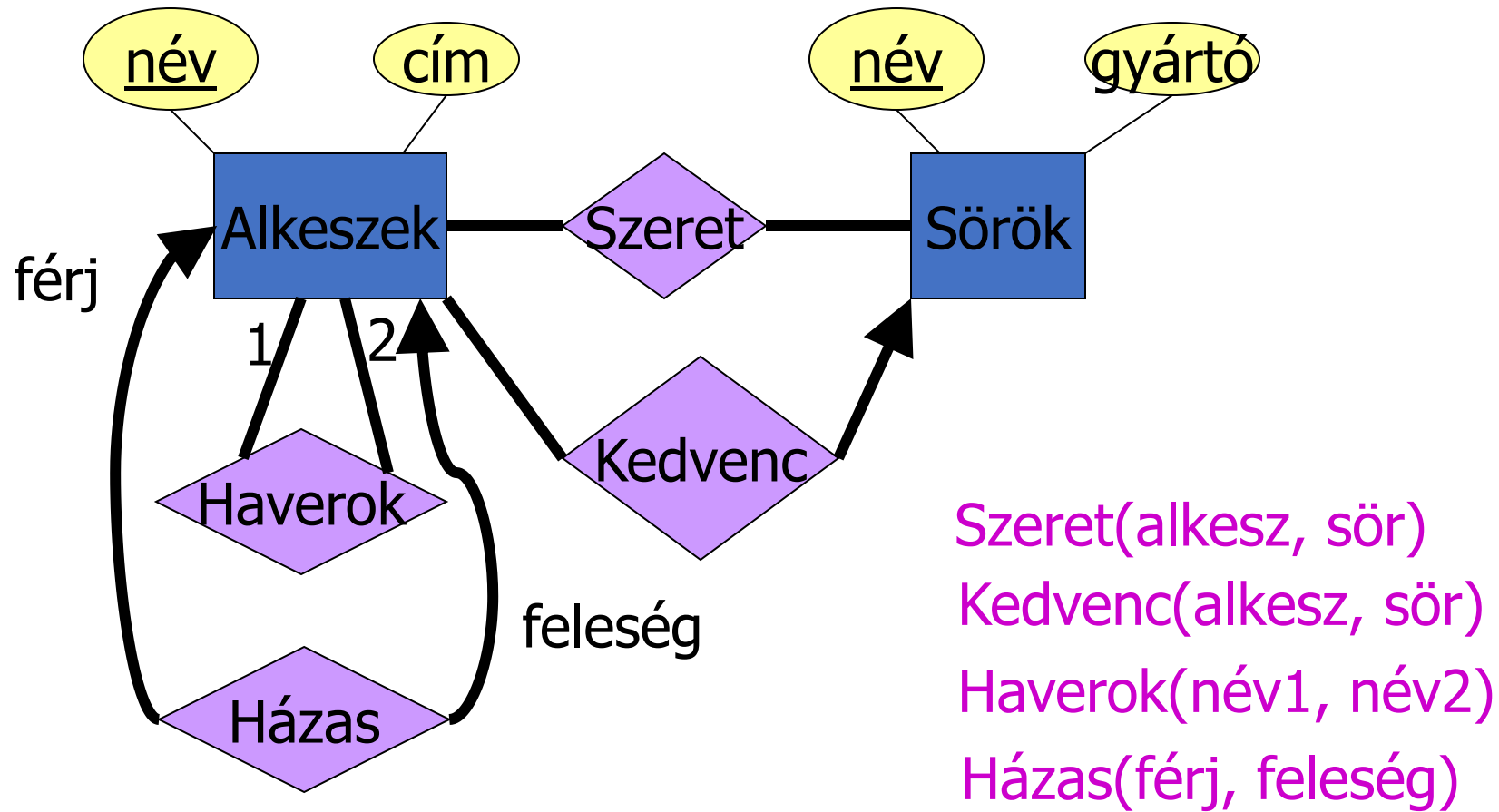
- Egyedhalmaz -> reláció.
  - Attribútumok -> attribútumok.
- Kapcsolat -> relációk, melyeknek az attribútumai csak:
  - az összekapcsolt egyedhalmazok kulcs-attribútumait,
  - és a kapcsolat attribútumait tartalmazzák.

# Egyedhalmaz -> Reláció



Reláció: Sörök(név, gyártó)

# Kapcsolat -> Reláció





# Relációk összevonása

- Egy relációba összevonhatók:
  1. Az  $E$  egyedhalmazból kapott reláció,
  2. valamint azon sok-egy kapcsolatok relációi, melyeknél az  $E$  a „sok” oldalon szerepel.
- **Példa:**  $\text{Alkeszek}(\text{név}, \text{cím})$  és  $\text{Kedvenc}(\text{alkesz}, \text{sör})$  összevonható az  $\text{Alkeszek1}(\text{név}, \text{cím}, \text{kedvencSör})$  relációvá.

# A sok-sok kapcsolat nem összevonható

- Az Alkeszek és Szeret relációk összevonása hiba lenne, redundanciához vezet.

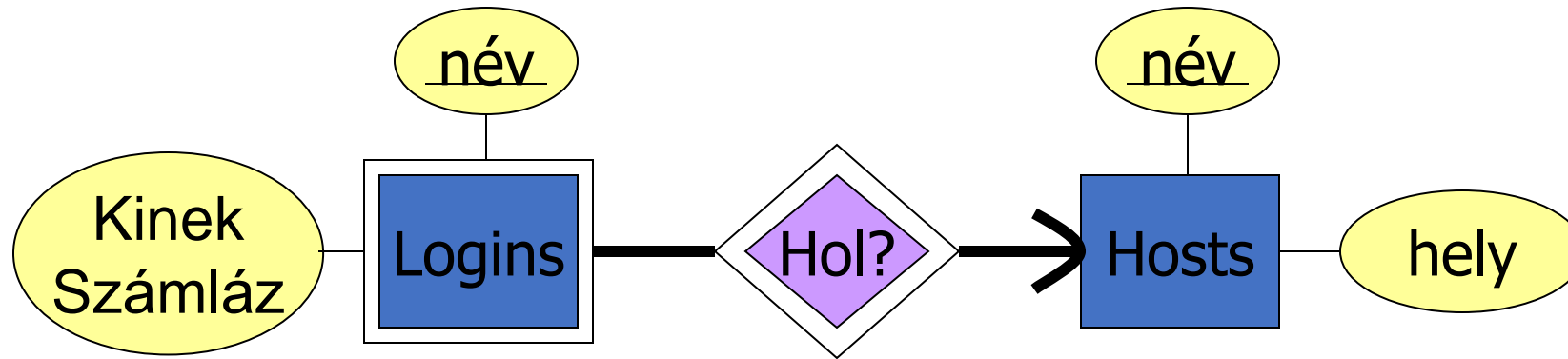
név	cím	Sör
Susy	123 Maple	Bud
Susy	123 Maple	Miller

Redundancia

# Gyenge egyedhalmazok átírása

- Egy gyenge egyedhalmazokból kapott relációnak a teljes kulcsot tartalmaznia kell (a más egyedhalmazokhoz tartozó kulcs-attribútumokat is), valamint a saját, további attribútumokat.
- A támogató kapcsolatot nem írjuk át, redundanciához vezetne.

## Példa: gyenge egyedhalmaz -> reláció



Hosts(hostNév, hely) (Gazda gép, Kiszolgáló)

Logins(loginNév, hostNév, billTo)

~~Hol(loginNév, hostNév, hostNév2)~~

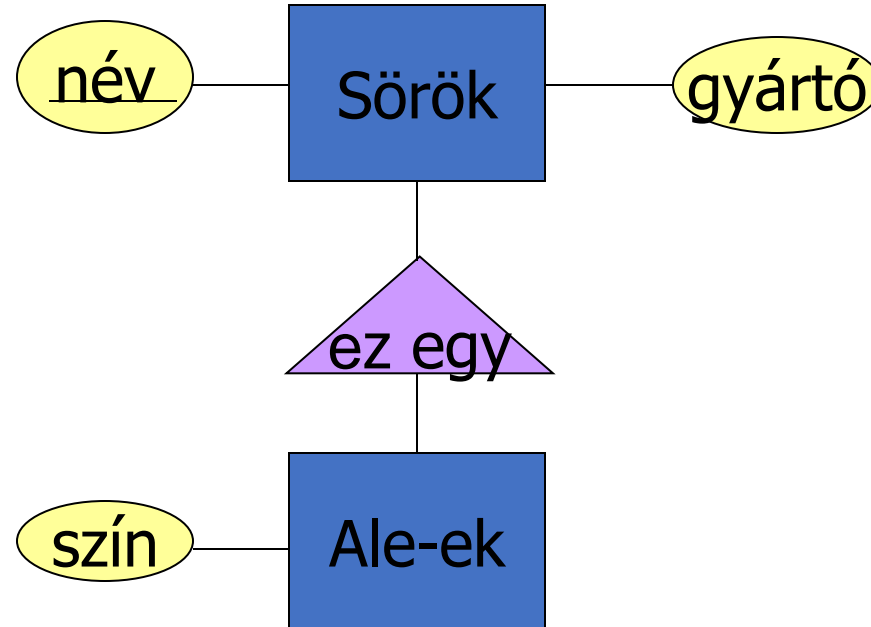
A Hol a Logins  
részévé válik.

Meg kell egyezniük.

# ***Egyed Alosztályok:*** három megközelítés

1. *Objektumorientált*
2. *Null értékek használata*
3. *E/K style*

## Példa: alosztály -> reláció



# Objektumorientált

név	gyártó
Bud	Anheuser-Busch

Sörök

név	gyártó	szín
Summerbrew	Pete's	barna

Ale-ek

A „Pete által gyártott ale-ek színeit” visszaadó  
lekérdezések esetén hasznos

# E/K Style

név	gyártó
Bud Summerbrew	Anheuser-Busch Pete's

Sörök

név	szín
Summerbrew	barna

Ale-k

A Pete által gyártott söröket visszaadó  
lekérdezések esetén hasznos.



# Null értékek használata

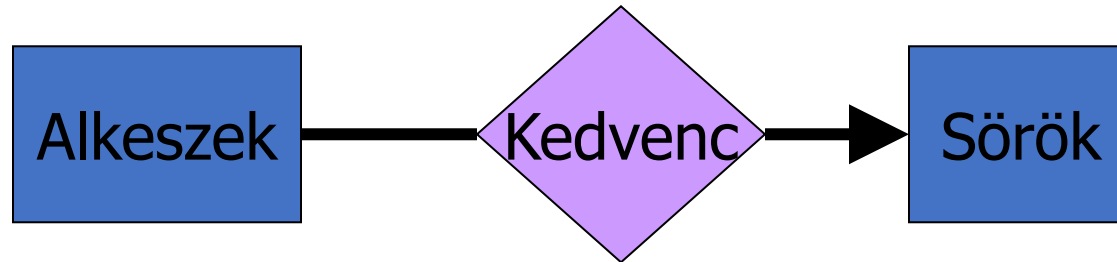
név	gyártó	szín
Bud Summerbrew	Anheuser-Busch Pete's	NULL barna

Sörök

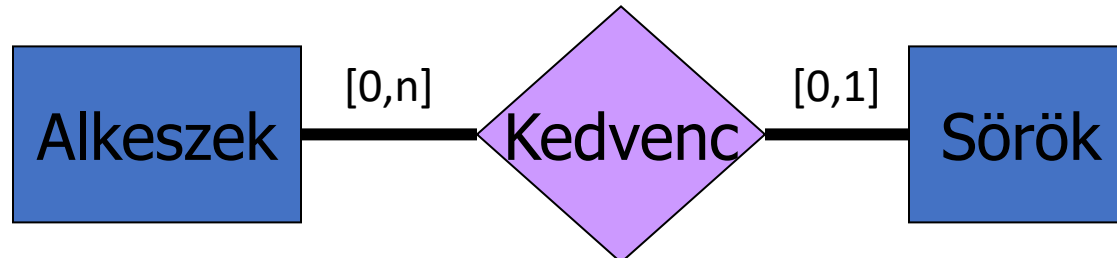
Helytakarékos, hacsak nem túl sok a NULL érték. Összekapcsolásokat is megspórolhatunk.

## Példa: sok-egy kapcsolat, jelölések

- Az **Alkeszekből** a **Sörök**be menő **Kedvenc** kapcsolat sok-egy típusú.
- Egy alkesznek legfeljebb egy kedvenc söre lehet.
- De egy sör több vagy akár nulla alkesznek is a kedvence lehet.

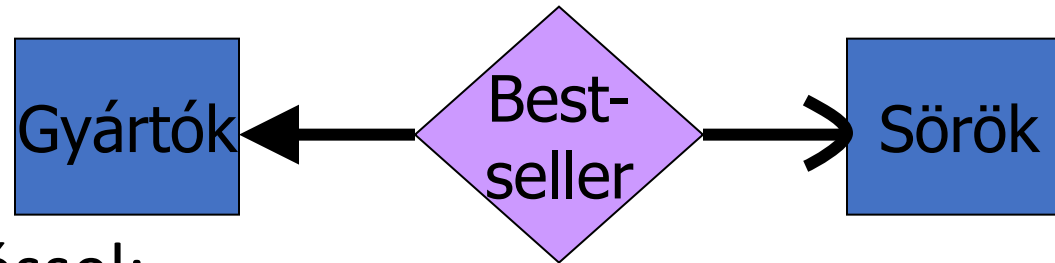


- Másik jelöléssel:



## Példa: egy-egy kapcsolat, jelölések

- Vegyük a Gyártók és Sörök közötti Bestseller kapcsolatot.
- Néhány sör valószínűleg egyetlen gyártónak sem lesz bestsellere, tehát a Gyártók egyedhalmaz felőli nyíl nem lehet kerek.
- De egy gyártónak kell lennie bestsellerének.



- Másik jelöléssel:

