

# Magas szintű adatbázismodellek

Tankönyv: Ullman-Widom:  
Adatbázisrendszerek Alapvetés  
Második, átdolgozott kiadás,  
Panem, 2009

---

4.1.- 4.4. E/K-modell elemei

4.5.- 4.6. E/K-diagram átírása  
relációs modellé

--- Kiegészítő tananyag:

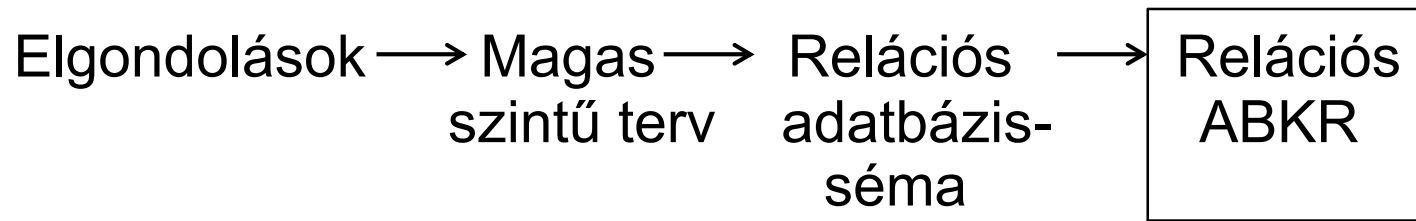
--- 4.7.- 4.8. UML-diagram átírása relációs modellé

--- (nincs a vizsgán, de a szakdolgozatban hasznos)



# Magas szintű adatbázismodellek

- Vizsgáljuk meg azt a folyamatot, amikor egy új adatbázist létrehozunk, vegyük példaként a sörivós adatbázist.
- Az adatbázis-modellezés és implementálás eljárása



- Modellezés
  - komplex valós világ leképezése, absztrakció
- Tervezési fázis:
  - Milyen információkat kell tárolni?
  - Mely információelemek kapcsolódnak egymáshoz?
  - Milyen megszorításokat kell figyelembe venni? stb...

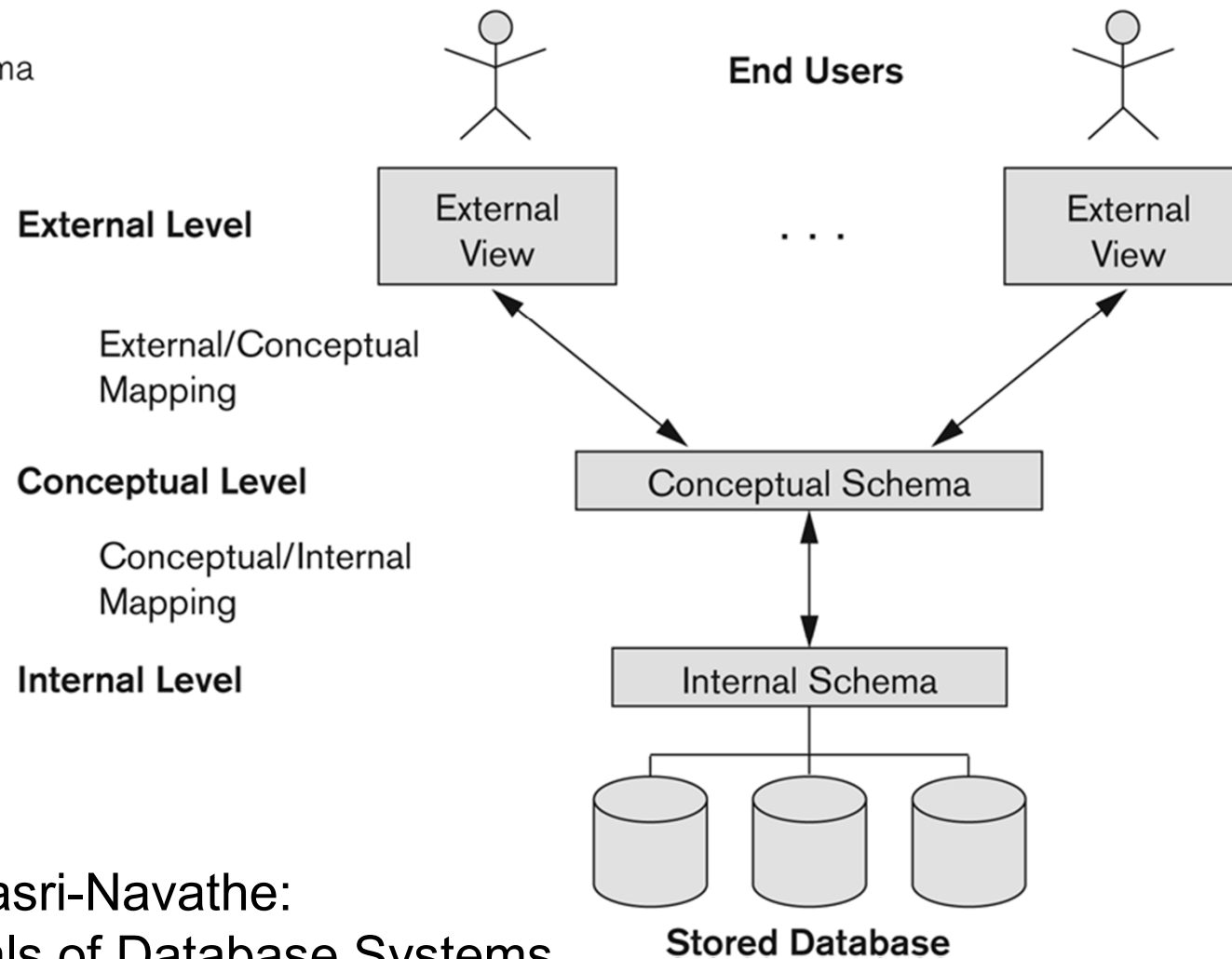
# Az adatmodellek 3 szintje

- Hogyan látjuk az adatbázist?
- A 3 szintű ANSI/SPARC architektúra
  - **Logikai** (külső, a felhasználói szemléletnek megfelelő szinten, nézetek)
  - **Fogalmi** (conceptual) (absztrakt, szintetizálja az összes felhasználói szemléletet)
  - **Fizikai** (belső, az adatbázis valamilyen fizikai adatstruktúrában letárolva a háttértárolón)

# Az adatmodellek 3 szintje

**Figure 2.2**

The three-schema architecture.



Forrás: Elmasri-Navathe:  
Fundamentals of Database Systems

# Az adatbázis-sématervezés lépései

Relációs adatbázisok tervezésének elméletére később a félév végén visszatérünk (Tk.3.fejezete)

- **Top-down:** a feldolgozandó információ elemzése,
- az információk közti **kapcsolatok meghatározása**
- az eredmény ábrázolása (**E/K diagram**),
- **adatbázisterv készítése** (transzformációs lépés),
- adatbázisterv finomítása (**összevonások**),
- **Bottom-up:** megszorítások modellezése, (funkcionális, többértékű) **függőségek meghatározása**
- relációs adatbázissémák tervezése, **dekomponálás:** (VM, FŐ felbontás), **normalizálás** (3NF, BCNF, 4NF),
- **Összevetés:** optimális adatbázisterv készítése,
- **Implementálás:** az **adatbázisterv megvalósítása.**

# Egyed-kapcsolat modell elemei

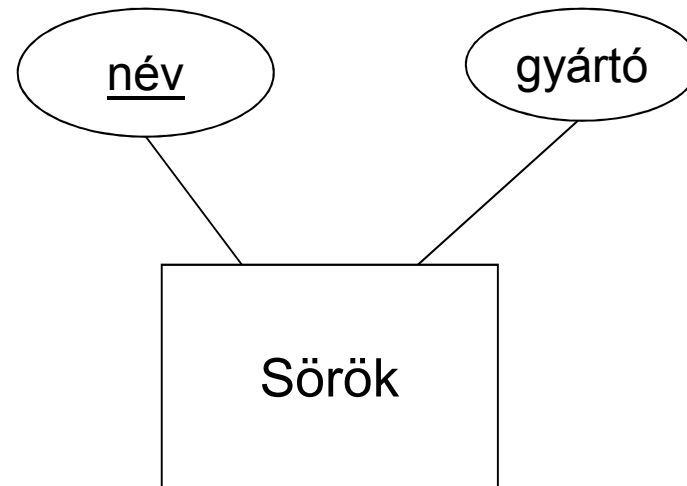
- **Egyed-kapcsolat modell: E/K modell**  
(Entity-relationship ER) alapfogalmak:
- **Egyedhalmazok** (absztrakt objektumok osztálya)
  - Miről gyűjtünk adatokat?
  - Mit tegyünk egy gyűjteménybe? - hasonlóság
  - Hasonló egyedek összessége
- **Attribútumok**
  - Megfigyelhető tulajdonságok, megfigyelt értékek
  - Az egyedek tulajdonságait írják le
- **Kapcsolatok**
  - Más egyedhalmazokkal való kapcsolatuk

# E/K modell elemei: Egyedhalmazok

- $E(A_1, \dots, A_n)$  egyedhalmaz **séma**:
  - E az egyedhalmaz neve,
  - $A_1, \dots, A_n$  tulajdonságok,
  - $\text{DOM}(A_i)$  – lehetséges értékek halmaza.
  - például: tanár(név, tanszék).
- $E(A_1, \dots, A_n)$  sémájú egyedhalmaz **előfordulása**:
  - A konkrét egyedekből áll
  - $E = \{e_1, \dots, e_m\}$  egyedek (entitások) halmaza, ahol
    - $e_i(k) \in \text{DOM}(A_k)$ ,
    - semelyik két egyed sem egyezik meg minden attribútumban (léteznek és megkülönböztethetők)

# E/K-diagram: Egyedhalmazok

- E/K diagram: séma-szinten grafikusan ábrázoljuk
- Egyedhalmazok: **téglalap**
- Tulajdonságok: **ovális**
- az elsődleges kulcshoz tartozó tulajdonságokat aláhúzzuk.



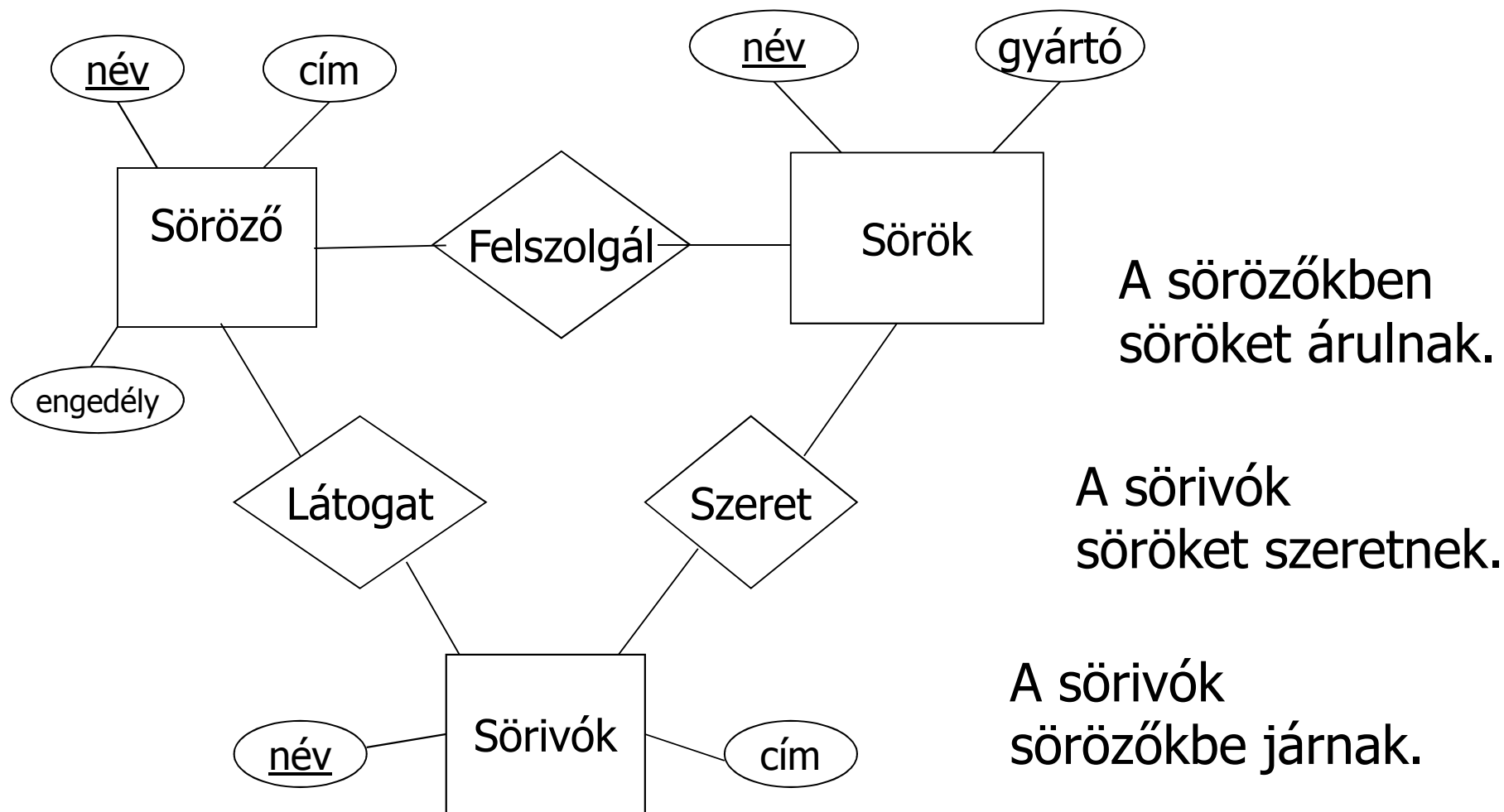


# E/K modell elemei: Kapcsolatok

- $K(E_1, \dots, E_p)$  a kapcsolat sémája,
  - $K$  a kapcsolat neve,
  - $E_1, \dots, E_p$  egyedhalmazok sémái,
  - $p=2$  bináris kapcsolat,  $p>2$  többágú kapcsolat,
  - például: tanít(tanár, tárgy).
- $K(E_1, \dots, E_p)$  sémájú kapcsolat előfordulása:
  - $K = \{(e_1, \dots, e_p)\}$  egyed  $p$ -esek halmaza, ahol
    - $e_i \in E_i$ ,
    - a kapcsolat előfordulásaira tett megszorítások határozzák meg a kapcsolat típusát.

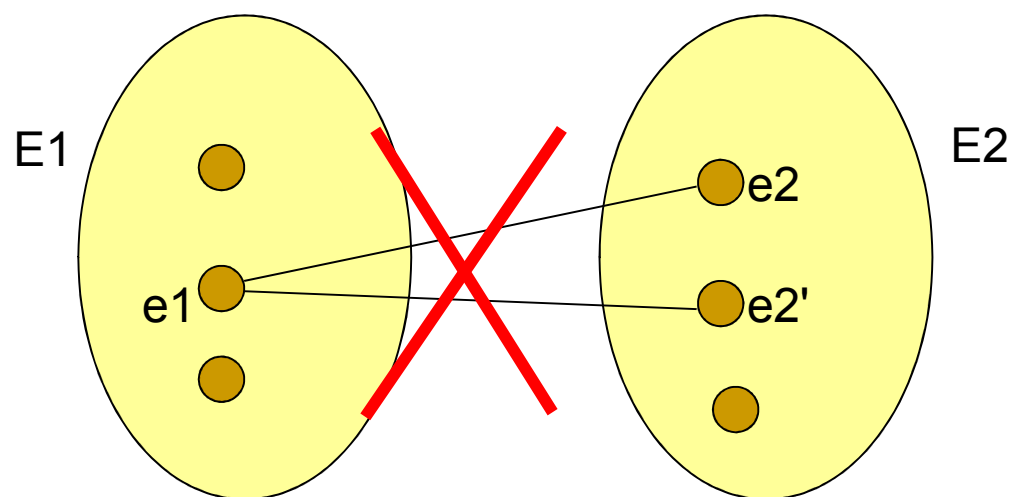
# E/K-diagram: Kapcsolatok

➤ A kapcsolatok jele: **rombusz**



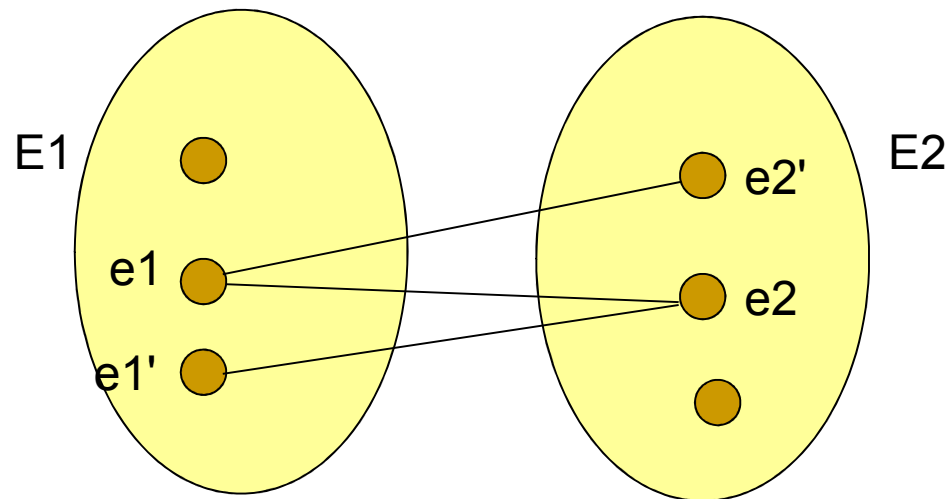
# Kapcsolatok típusai

- $K(E1, E2)$  bináris kapcsolat, sok-egy (n:1)
  - $K \{(e_i, e_j)\}$  alakú előfordulásaiban nem szerepelhet egyszerre  $(e1, e2)$  és  $(e1, e2')$ , ha  $e2$  és  $e2'$  különböznek,
  - másképpen:  **$K$  előfordulásaiban minden  $E1$ -beli egyedhez legfeljebb 1  $E2$ -beli egyed tartozhat,**
  - például: született(név, ország).



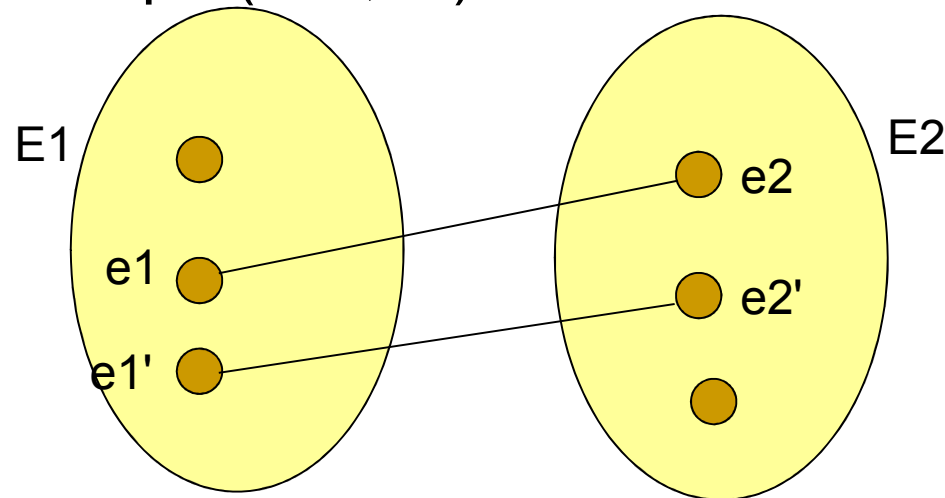
# Kapcsolatok típusai

- $K(E1, E2)$  bináris kapcsolat, **sok-sok** (n:m),
  - $K \{(e_i, e_j)\}$  alakú előfordulásai nincsenek korlátozva,
  - előfordulhat (de nem kötelező, hogy előforduljon) az ábrán látható helyzet, vagyis **minden E1-beli egyedhez több E2-beli egyed tartozhat, és fordítva, minden E2-beli egyedhez több E1-beli egyed tartozhat,**
  - például: tanul(diák, nyelv).

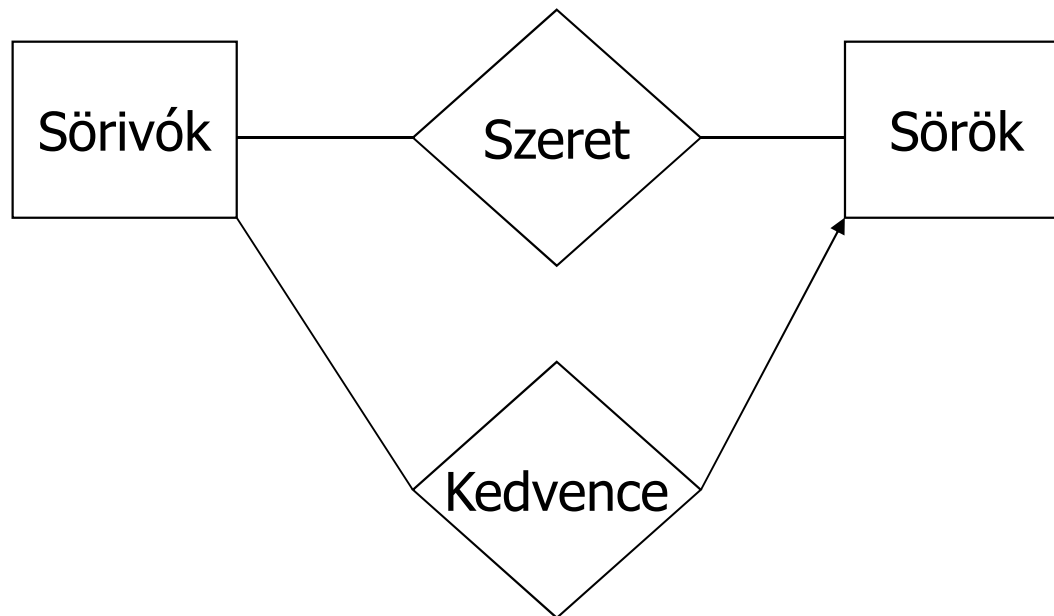


# Kapcsolatok típusai

- $K(E1, E2)$  bináris kapcsolat, egy-egy (1:1),
- $K \{(e_i, e_j)\}$  alakú előfordulásai egyszerre sok-egy és egy sok típusúak, vagyis minden  $E1$ -beli egyedhez legfeljebb egy  $E2$ -beli egyed tartozhat, és fordítva, minden  $E2$ -beli egyedhez legfeljebb egy  $E1$ -beli egyed tartozhat,
- nem kötelezően szerepel minden egyed a kapcsolatban,
- például: házaspár(férfi,nő).



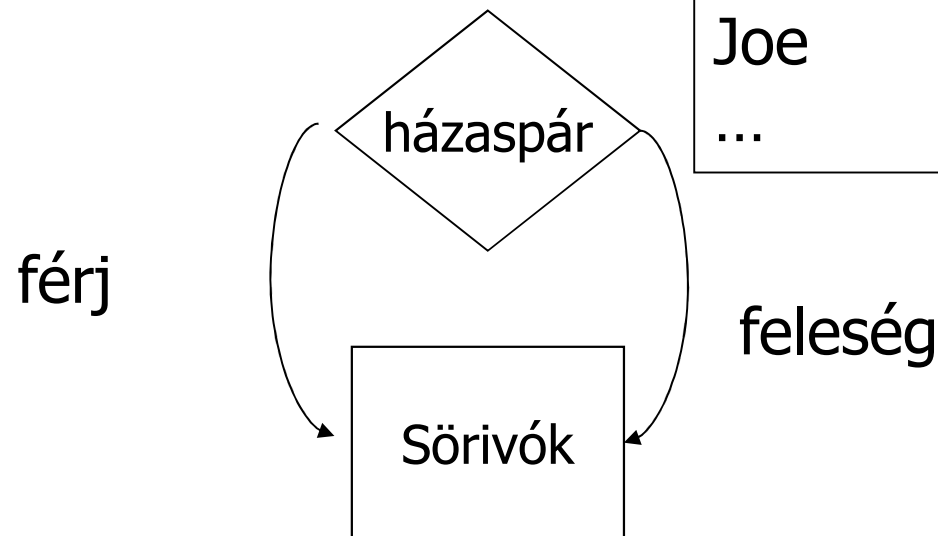
# Két egyedhalmaz között több kapcsolat is lehet



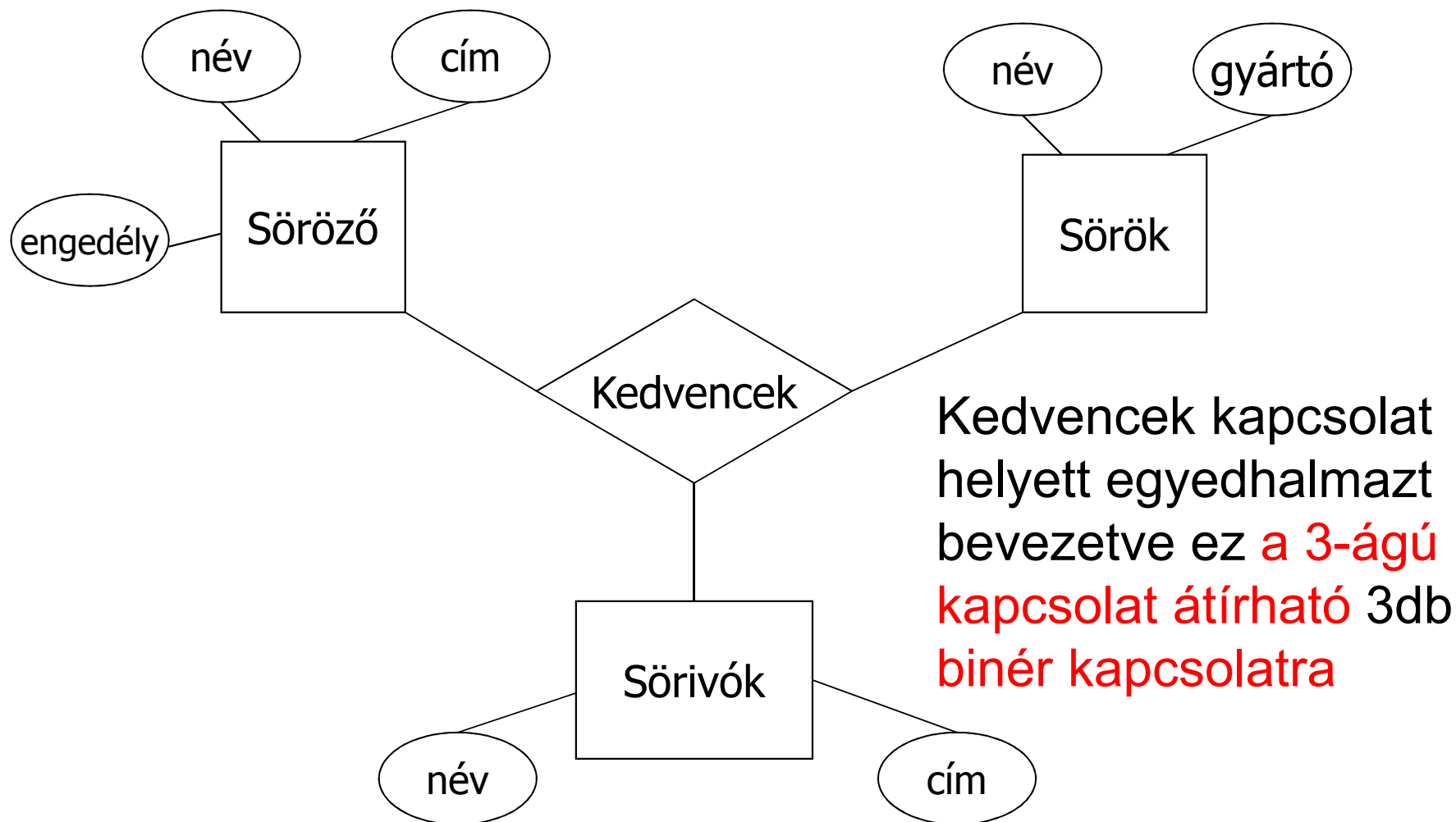
# Egy egyedhalmaz önmagával is kapcsolódhat: Szerepek (Roles)

A kapcsolat előfordulása

Férj	Feleség
Bob	Ann
Joe	Sue
...	...

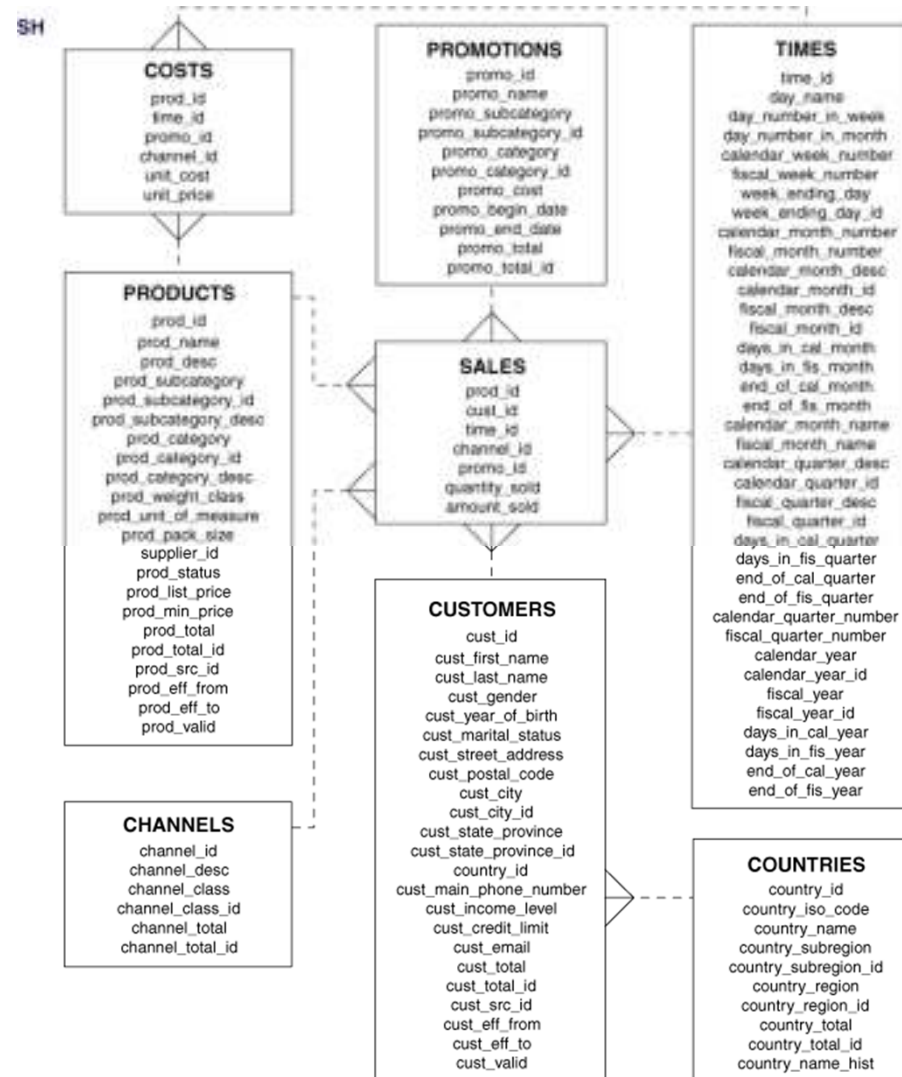


# Példa: Többágú (3-ágú) kapcsolatra

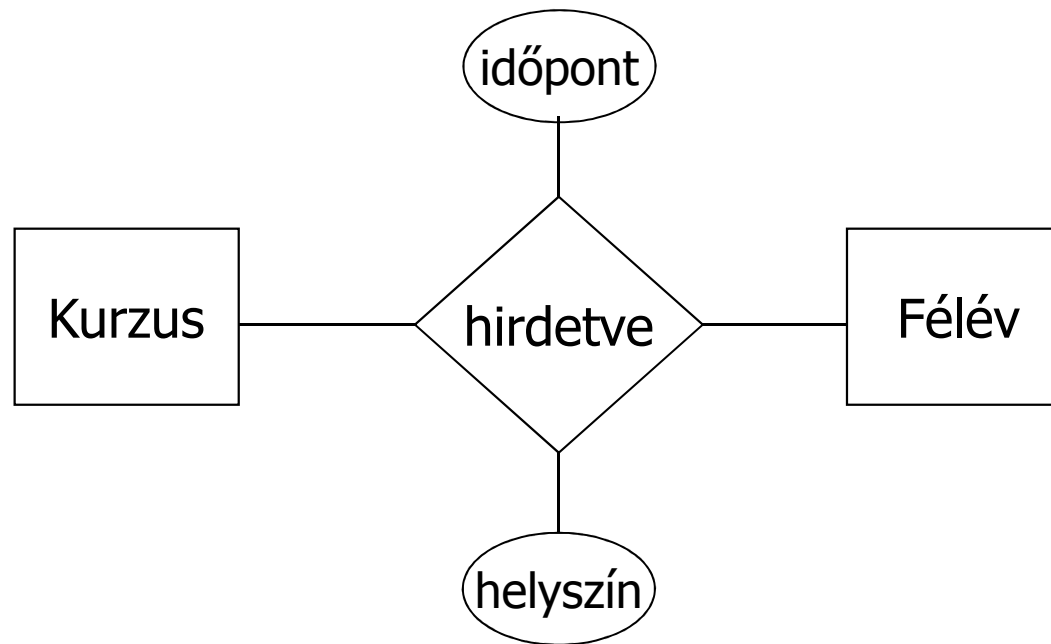




# Példa: Oracle SH (Sales History) séma

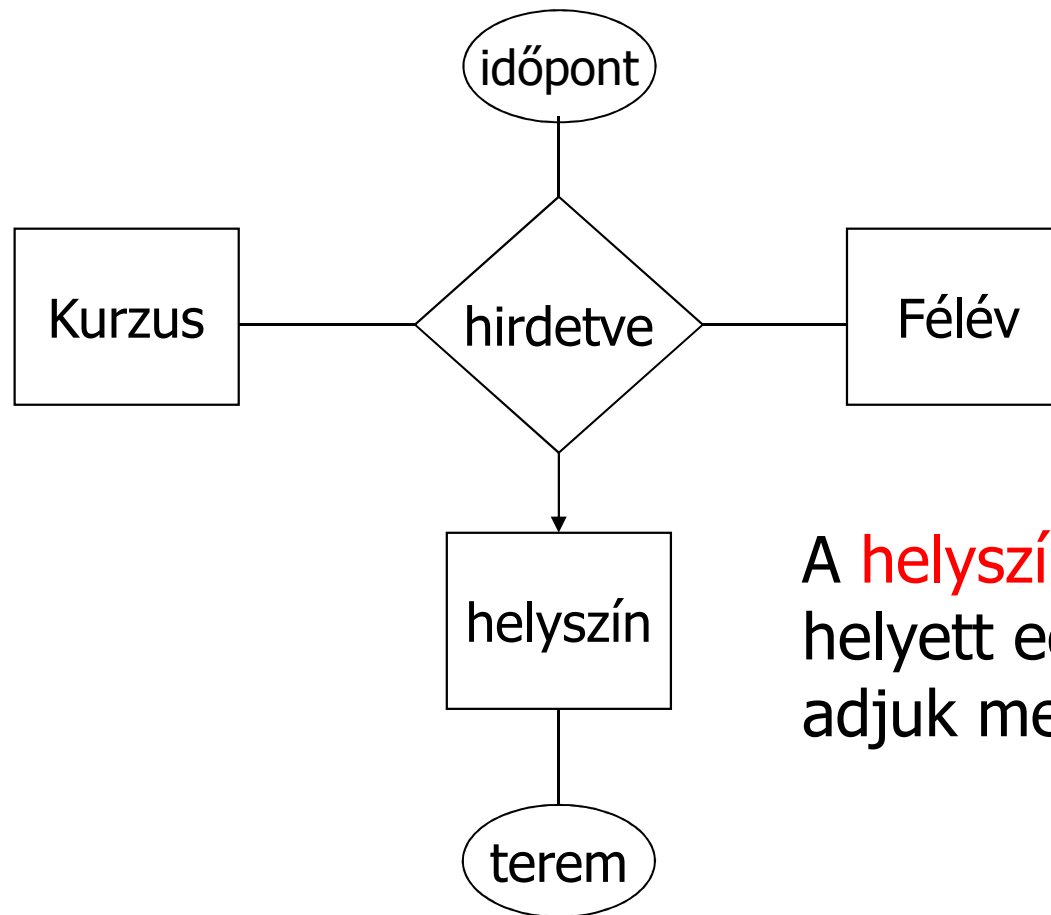


# Kapcsolatnak is lehet attribútuma



Az **időpont** és **helyszín** a Kurzus és Félév együttes függvénye, de egyiké sem külön.

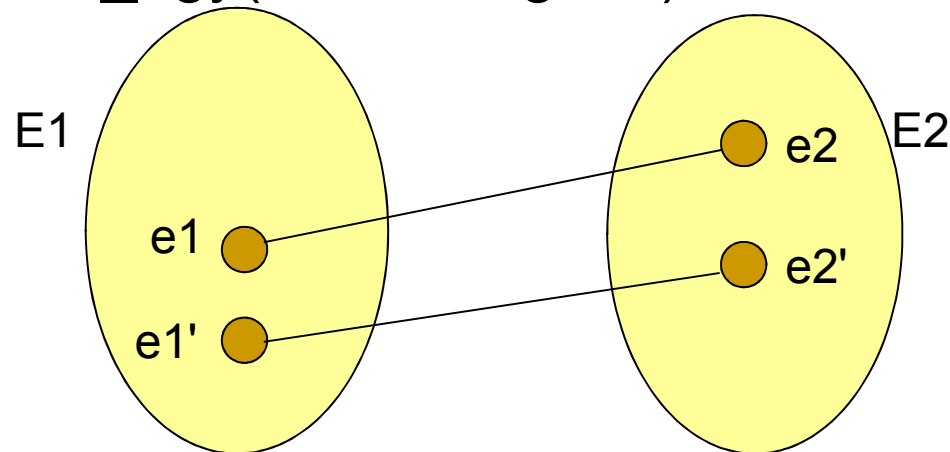
# Tervezési kérdés: Attribútum vagy egyedhalmaz?



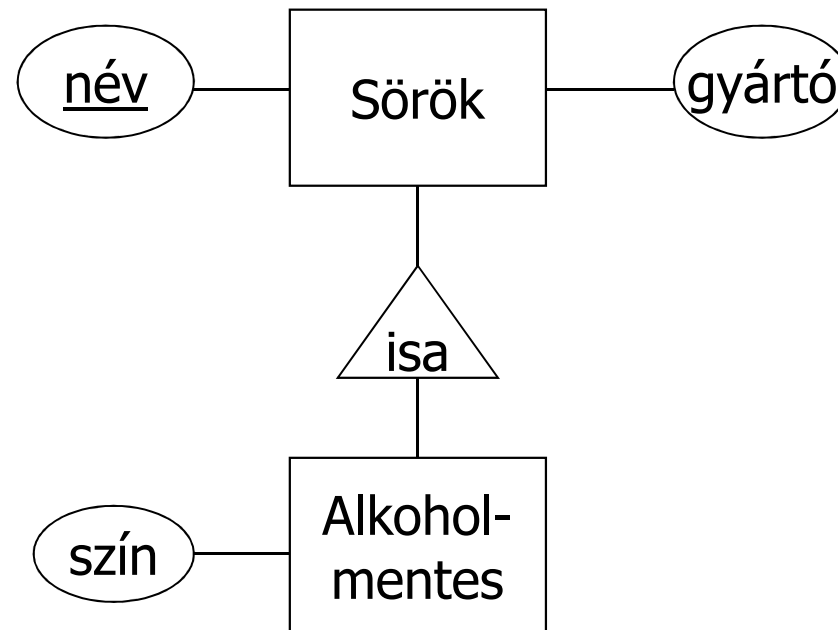
A **helyszínt** itt attribútum helyett egyedhalmazként adjuk meg

# Speciális „is-a” (az-egy) kapcsolat

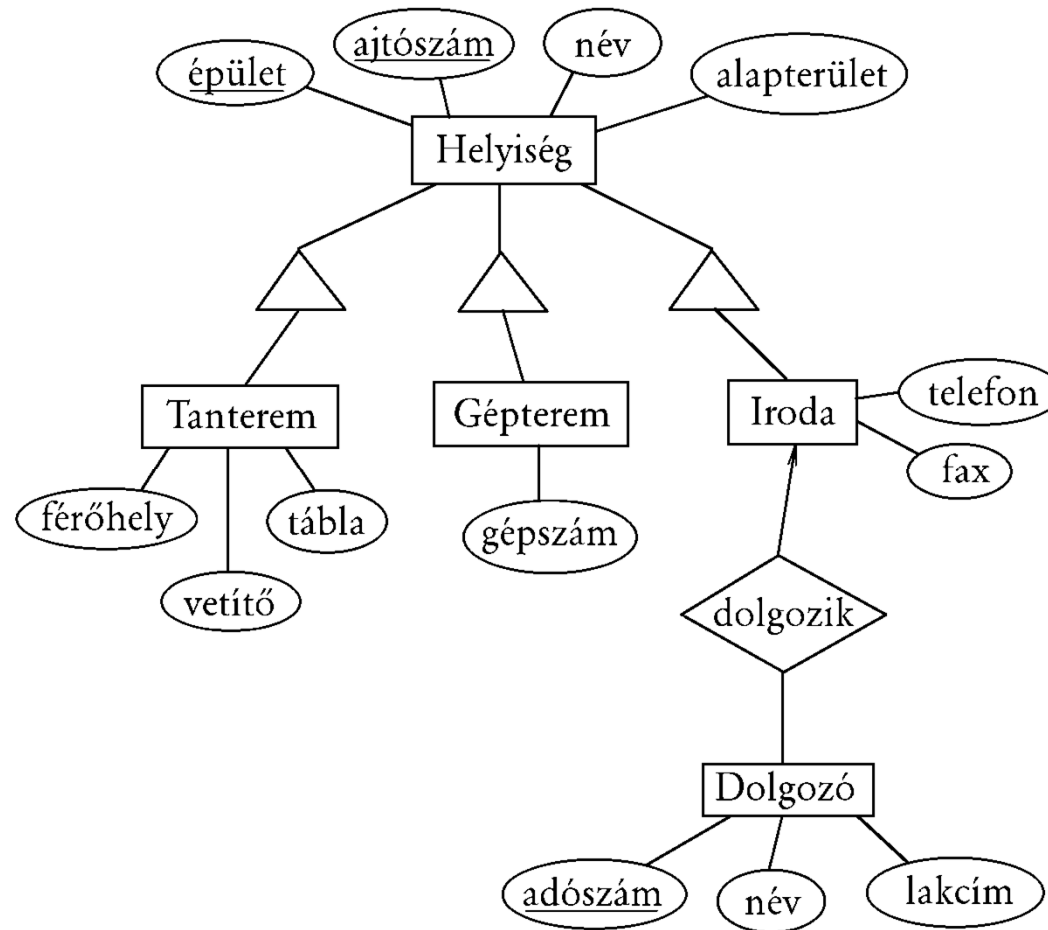
- $K(E1, E2)$  bináris kapcsolat,
  - öröklődési kapcsolat ("az egy", ISA),
  - "a PC is a computer" = "a PC az egy számítógép",
  - speciális egy-egy kapcsolat,
  - $K \{(e_i, e_j)\}$  alakú előfordulásaiban az összes E1-beli egyed szerepel,
  - például: az\_egy(főnök, dolgozó).



# Alosztályok és öröklődés



# Példa: „is-a” (az-egy) kapcsolatra



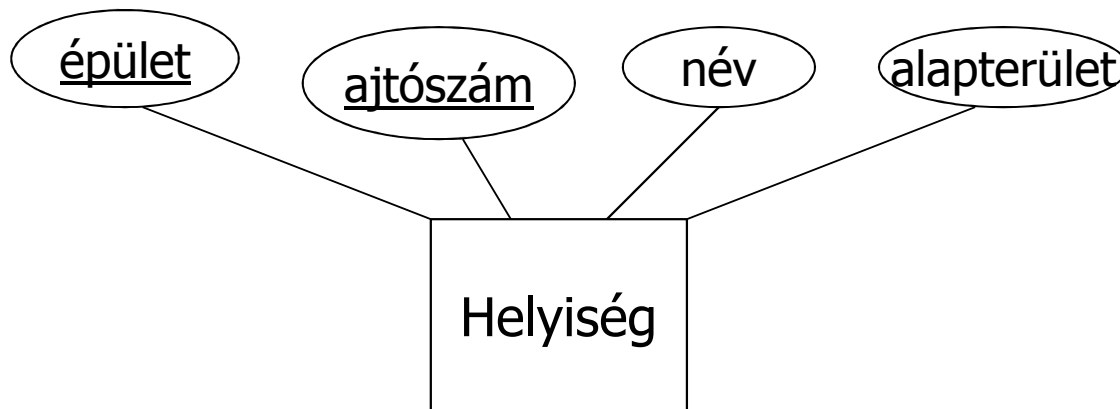
# Kulcs megszorítás

## jele: aláhúzás

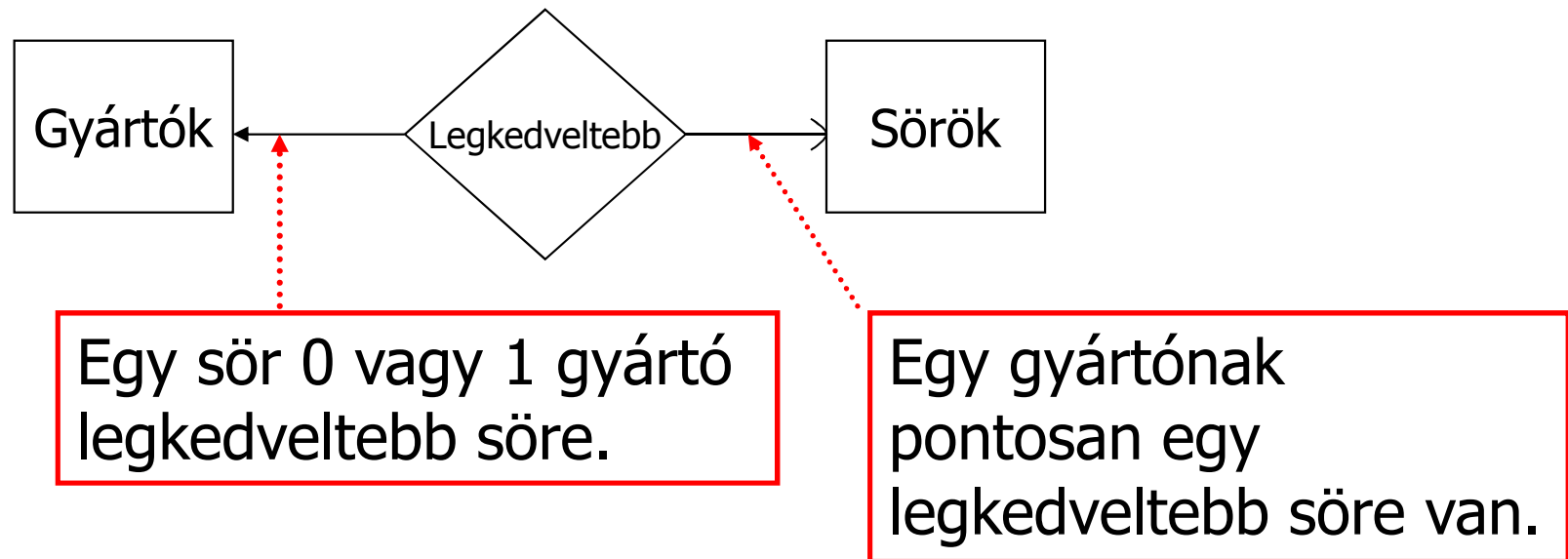
Példa egyszerű kulcsra: név a Sörök elsődleges kulcsa:



Példa összetett kulcsra: épület, ajtószám két-attribútumos elsődleges kulcsa a Helyiség-nek:

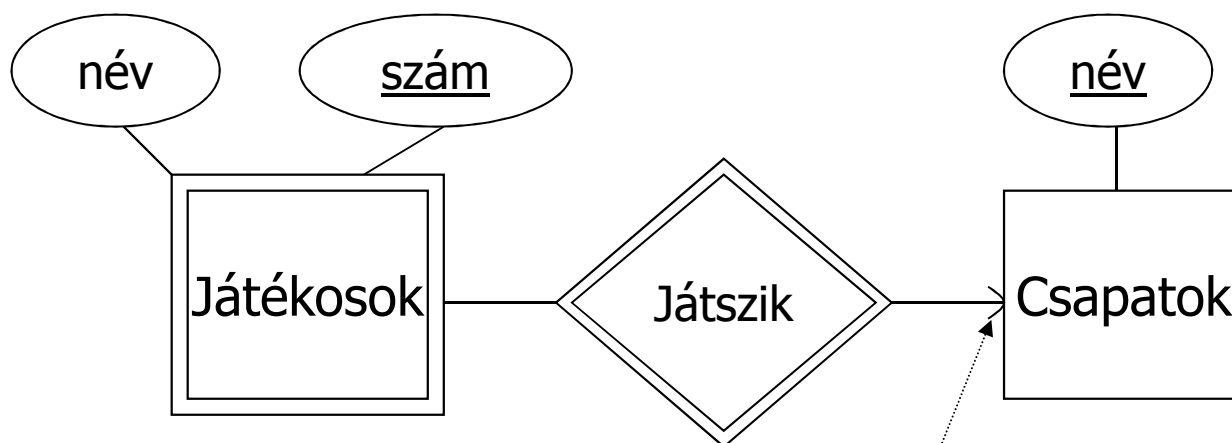


# Hivatkozási épség megszorítás jele a kerek végződés —)





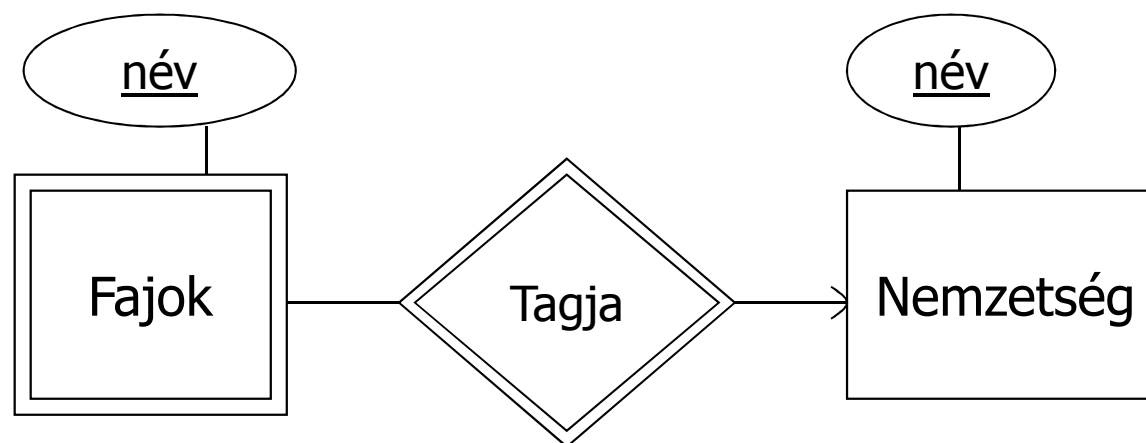
# Erős és gyenge egyedhalmaz



A kerek végződés jelzi, hogy minden játékoshoz kötelezően tartozik egy csapat, amely az azonosításhoz használható.

- Dupla rombusz: sok-egy gyenge kapcsolat.
- Dupla téglalap: gyenge egyedhalmaz.

# Erős és gyenge egyedhalmaz



Tankönyv 4.21. példája:  
például az emberek a **Homo sapiens** fajhoz tartoznak,  
ahol **Homo** a nemzetség neve, a **sapiens** a faj neve  
(sajnos maguk a fajok nevei nem egyértelműek, két  
vagy több nemzetségben is lehet ugyanolyan fajnév).

# Tervezési alapelvek

- **valósághű modellezés:**
  - megfelelő tulajdonságok tartozzanak az egyedosztályokhoz, például a tanár neve ne a diák tulajdonságai közé tartozzon
- **redundancia elkerülése:**
  - az `index(etr_kód,lakcím,tárgy,dátum,jegy)` **rossz séma**, mert a lakcím annyiszor ismétlődik, ahány vizsgajegye van a diáknak, helyette 2 sémát érdemes felvenni:  
`hallgató(etr_kód,lakcím)`, `vizsga(etr-kód,tárgy,dátum,jegy)`.
- **egyszerűség:**
  - fölöslegesen ne vegyünk fel egyedosztályokat
  - például a `naptár(év,hónap,nap)` helyett a megfelelő helyen inkább `dátum` tulajdonságot használjunk
- **tulajdonság vagy egyedosztály:**
  - például a `vizsgajegy` osztály helyett `jegy` tulajdonságot használjunk.

# Modellezési feladatok (Tankönyv)

- **4.1.1. feladat.** Tervezzünk egy bank részére adatbázist, amely tartalmazza az ügyfeleket és azok számláit. Az ügyfelekről tartsuk nyilván a nevüket, címüket, telefonszámukat és TAJ-számukat. A számláknak legyen számlaszámuk, típusuk (pl. takarékbetét-számla, folyószámla stb.) és egyenlegük. Továbbá, meg kell jelölni azokat az ügyfeleket, akiknek van számlájuk. Adjuk meg az E/K diagramját ennek az adatbázisnak. Alkalmazzunk nyilakat a kapcsolatokban a multiplicitások jelölésére.

# Modellezési feladatok (Tankönyv)

- **4.1.3. feladat.** Adjuk meg az E/K modelljét egy olyan adatbázisnak, amely csapatokat, játékosokat és azok szurkolóit tartja nyilván:
  - Minden csapatról tároljuk a nevét, játékosait, csapatkapitányát (ő is egy játékos), mezük színét.
  - Minden játékosnak legyen neve.
  - Minden rajongóról tartsuk nyilván a nevét, kedvenc csapatát, kedvenc játékosát és kedvenc színét.
- Vigyázzunk, a színek halmaza nem lehet a csapatok egy attribútumának típusa. Hogyan lehet ezzel a megszorítással együtt megfelelő modellt készíteni?

# Modellezési feladatok (Tankönyv)

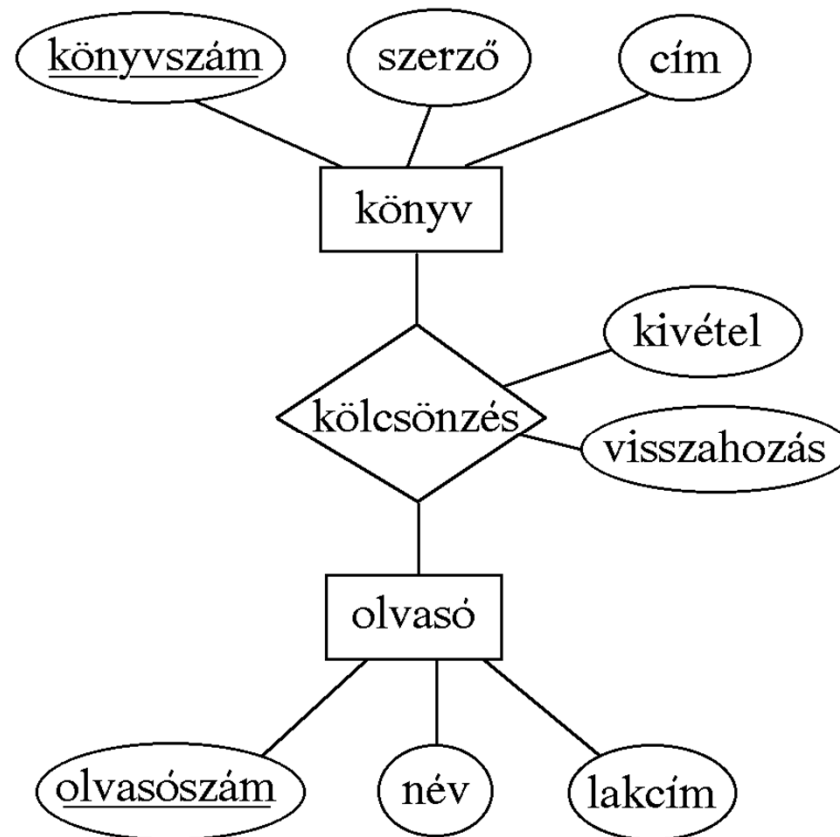
- **4.1.9. feladat.** Tervezzünk adatbázist egy tanulmányi osztály számára. Ez az adatbázis tartalmazza a hallgatókat, oktatókat, tanszékeket és kurzusokat. Ezenkívül tartsuk nyilván, hogy a hallgatók milyen kurzusokat vettek fel, az adott kurzust mely oktató oktatja, a hallgatók jegyeit, a kurzusoknál az oktató munkáját segítő hallgatókat, egy adott kurzust mely tanszék ajánlotta, és minden olyan információt, ami a fentiek megvalósításához szükséges. Megjegyezzük, hogy ez a feladat nagy szabadságot enged a korábbiakhoz képest. Dönteni kell a kapcsolatok típusáról (sok-sok, sok-egy vagy egy-egy), az alkalmas típus megválasztásról, illetve arról, hogy milyen segédinformációkat használunk.

# E/K-diagram átírása

## Tankönyv 4.5.-4.6. E/K-diagram átírása relációkká

- Egyedhalmazok átírása relációkká
- E/K-kapcsolatok átírása relációkká
- Egyszerűsítés, összevonások
- Gyenge egyedhalmazok kezelése
- Osztályhierarchia átalakítása relációkká

# Példa: Egy könyvtár adatmodellje

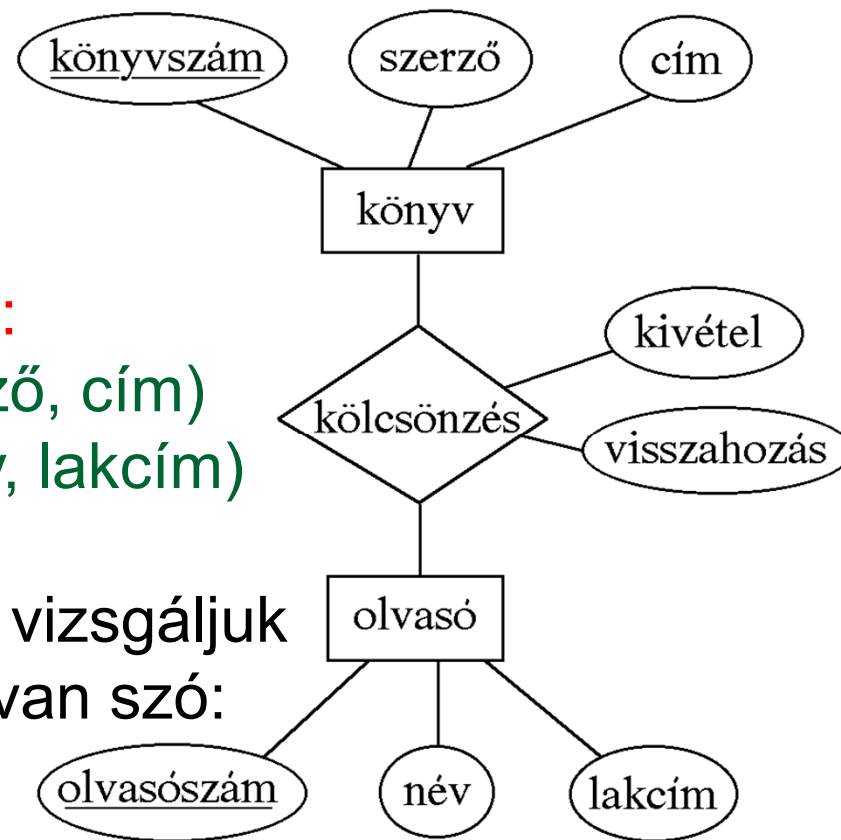




# E/K diagram átírásra példa

## 1.) Egyedhalmazok átírása:

KÖNYV (könyvszám, szerző, cím)  
OLVASÓ (olvasószám, név, lakcím)



Egyedhalmazok átírásánál vizsgáljuk meg, ha összetett típusról van szó: Például a lakcím (rekord) vagy a szerző (többértékű, halmaz vagy rendezett lista) milyen megoldások lehetnek?

## 2.) Ezután jön majd a kapcsolatok átírása:

KÖLCSÖN (könyvszám, olvasószám, kivétel, visszahozás)

# Egyedhalmazok átírásánál az összetett attribútumok leképezése

- Tegyük fel, hogy az OLVASÓ táblában a **lakcím** attribútumot (helység, utca, házszám) struktúraként, vagyis rekordként szeretnénk kezelni.
- Relációs adatmodellben erre egy lehetőség van: az OLVASÓ (olvasószám, név, lakcím) séma helyett a OLVASÓ (olvasószám, név, helység, utca, házszám) sémára térünk át.

# Egyedhalmazok relációkká való átírásánál

## Többértékű attribútumok leképezése ---1

- Kérdés, hogy többszerzős könyveket hogyan tartsunk nyilván az adatbázisban.
- **1.megoldás: Megadás egyértékű attribútumként.**  
A szerző megadására szolgáló szövegmezőben felsoroljuk a szerzőket.
- Hátrányok:
  - a szerzőket külön-külön nem tudjuk kezelni
  - sok szerző esetleg nem fér el a megadott mezőben

# Egyedhalmazok relációkká való átírásánál

## Többértékű attribútumok leképezése ---2

➤ 2.megoldás: Megadás többértékű attribútumként.

a.) Sorok többszörözése. A KÖNYV táblában egy könyvhöz annyi sort veszünk fel, ahány szerzője van:

Könyvszám	Szerző	Cím
1121	Ullman	Adatbázisok
1121	Widom	Adatbázisok
3655	Radó	Világatlasz
2276	Karinthy	Így írtok ti
1782	Jókai	Aranyember

A megfelelő relációséma:

KÖNYV (könyvszám, szerző, cím)

➤ A fenti megoldás hátránya, hogy a többszerzős könyvek címét több példányban kell megadni, ami redundanciát jelent.

# Egyedhalmazok relációkká való átírásánál

## Többértékű attribútumok leképezése ---3

b.) új tábla felvétele: KÖNYV (könyvszám, szerző, cím)  
sémát az alábbi két sémával helyettesítjük:

KÖNYV (könyvszám, cím)

SZERZŐ (könyvszám, szerző)

c) Sorszámozás. Ha a szerzők sorrendje nem közömbös, akkor a SZERZŐ táblát egy sorszám mezővel kell bővíteni (emlékeztetünk rá, hogy a relációs adatmodell nem definiálja a rekordok sorrendjét):

KÖNYV (könyvszám, cím)

SZERZŐ (könyvszám, sorszám, szerző)

# E/K diagram átírása relációs adatbázistervre

Mi minek felel meg:

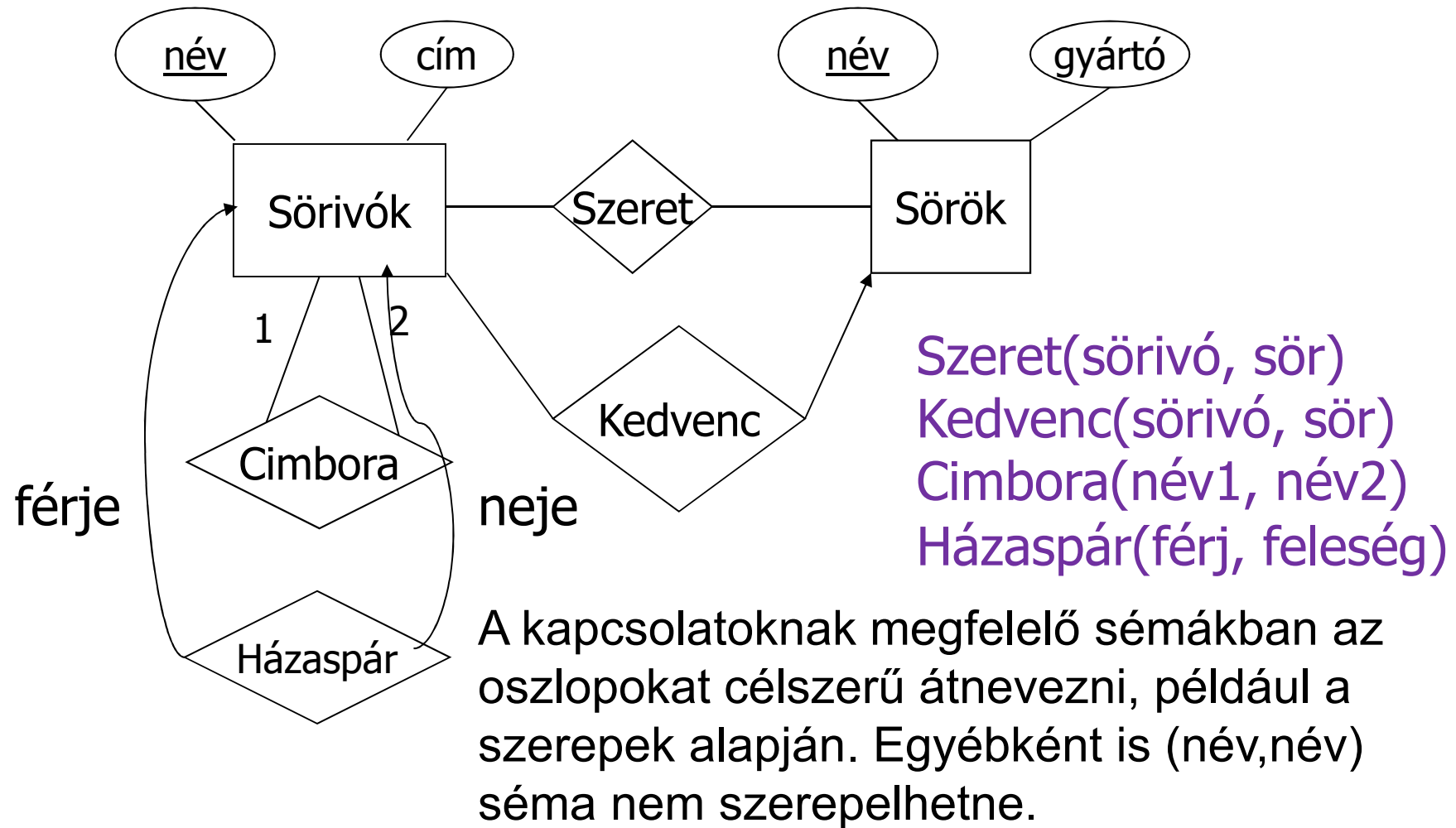
➤ egyedhalmaz séma	↔	relációséma
$E(A_1, \dots, A_n)$		$E(A_1, \dots, A_n)$
➤ tulajdonságok	↔	attribútumok
➤ (szuper)kulcs	↔	(szuper)kulcs
➤ egyedhalmaz előfordulása	↔	reláció
➤ e egyed	↔	$(e(A_1), \dots, e(A_n))$ sor
➤ $R(E_1, \dots, E_p, A_1, \dots, A_q)$	↔	$R(K_1, \dots, K_p, A_1, \dots, A_q)$
kapcsolati séma, ahol		relációséma, ahol
$E_i$ egyedhalmaz,	↔	$K_i$ az $E_i$ (szuper)kulcsa
$A_j$ saját tulajdonság		
E/K modell	↔	Relációs adatmodell

# E/K diagram átírása relációs adatbázistervre

- A transzformálás előtt a tulajdonságokat átnevezhetjük, hogy a **relációsémában ne szerepeljen kétszer ugyanaz az attribútum**.
- Az **az\_egy kapcsolat** esetén a speciális osztály saját attribútumaihoz hozzávesszük az általános osztály (szuper)kulcsát.
- Ha  $R(E1, E2)$  sok-egy kapcsolat, akkor  $R(K1, K2)$  relációsémának a  $K1$  szuperkulcsa lesz.
- A **gyenge entitás** relációsémáját bővíteni kell a meghatározó kapcsolat(ok)ban szereplő egyed(ek) kulcsával.

# Példa: E/K diagram átírása relációkká

## Az egyedek átírása után a kapcsolatok átírása:

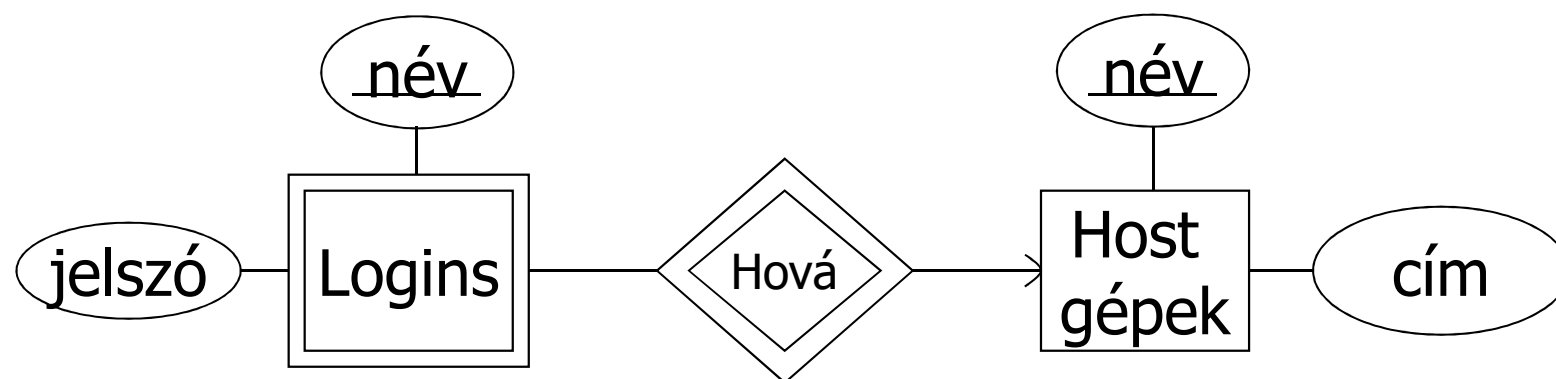




# Relációk összevonása

- Összevonhatunk 2 relációt, ha  
az egyik egy **sok-egy** kapcsolatnak megfelelő reláció, a másik pedig a sok oldalon álló egyedhalmaznak megfelelő reláció.
- **Példa:**  
Sörivók(név, cím) és Kedvenc(ivó,sör)  
összevonható, és kapjuk az  
Sörivó1(név,cím,kedvencSöre) sémát.

# Gyenge egyedhalmaz átírása



Hostgépek(hostNév, cím)

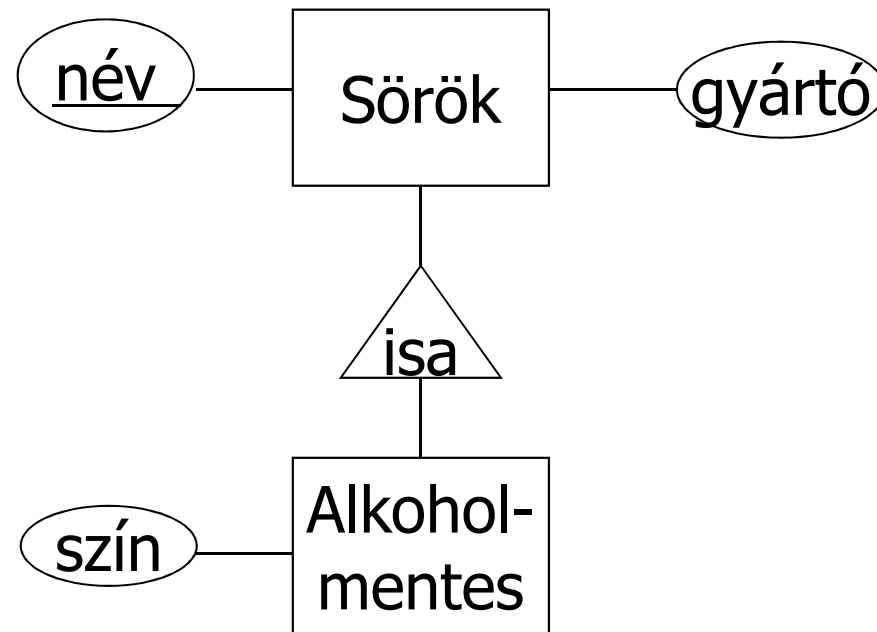
Logins(loginNév, hostNév, jelszó)

~~Hová(loginNév, hostNév, hostNév2)~~

Beolvastjuk a  
Logins relációba

A logins kulcsa összetett: loginNév, hostNév  
Kétszer szerepelne az azonos értékű  
hostNév a Hová sémában

# Alosztály átírására relációkká



# Alosztályok átírása: három megközelítés

- **E/R stílusban:** Egy reláció minden alosztályra, de az általános osztályból csak a kulcsokat vesszük hozzá a saját attribútumokhoz.
- **Objektumorientált stílusban:** Egy reláció minden alosztályra, felsorolva az összes tulajdonságot, beleértve az örökölteket is.
- **Nullértékek használatával:** Egyetlen reláció az öröklődésben résztvevő összes osztályra. Ha egy egyed nem rendelkezik egy alosztály speciális tulajdonságával, akkor ezt az attribútumot NULL értékkel töltjük majd ki.

# E/K típusú átalakítás ---1

név	gyártó
Bud Summerbrew	Anheuser-Busch Pete's

Sörök

név	szín
Summerbrew	világos

Alkoholmentes

Az olyan lekérdezésekre jó, hogy egy adott gyártó milyen söröket gyárt, beleértve az alkoholmenteseket is.

# Objektumorientált megközelítés ---2

név	gyártó
Bud	Anheuser-Busch

Sörök

név	gyártó	szín
Summerbrew	Pete's	világos

Alkoholmentes

Az olyan lekérdezésekre jó, hogy egy adott gyártó milyen színű alkoholmentes söröket gyárt.

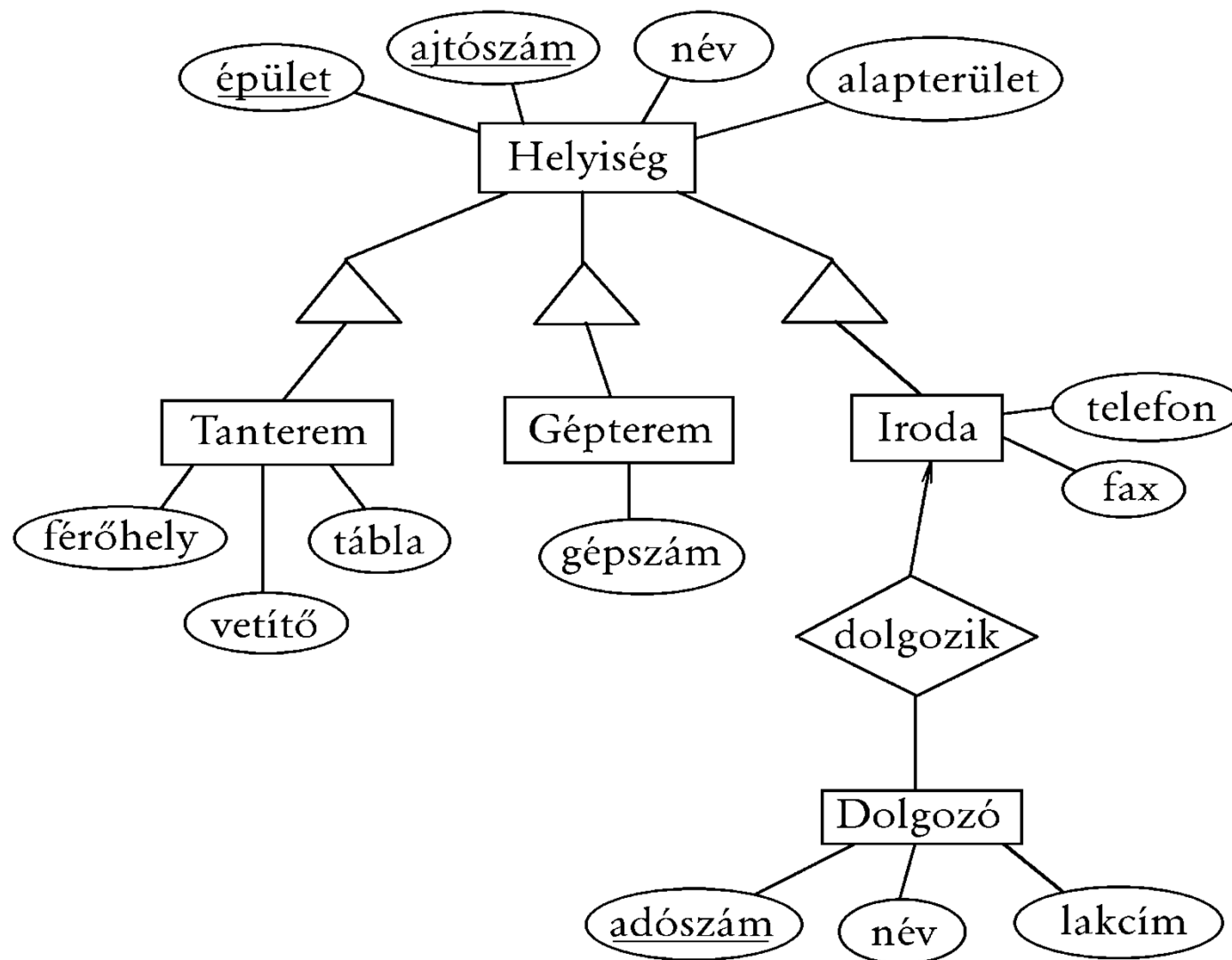
# Nullértékek használatával ---3

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

Sörök

Általában kevesebb hely elég a tárolásra, kivéve ha nagyon sok attribútum marad nullértékű.

# Példa: Alosztály átírása relációkká





# E/K típusú átalakítás --- 1

2. Minden altípushoz külön tábla felvétele, egy egyed több táblában is szerepelhet. A főtípus táblájában minden egyed szerepel, és annyi altípuséban ahánynak megfelel. Az altípusok a főtípustól csak a kulcs-attribútumokat öröklík. (E/K stílusú reprezentálás.)

HELYISÉG (épület, ajtószám, név, alapterület)

TANTEREM (épület, ajtószám, férőhely, tábla, vetítő)

GÉPTEREM (épület, ajtószám, gépszám)

IRODA (épület, ajtószám, telefon, fax)

DOLGOZÓ (adószám, név, lakcím, *épület, ajtószám*)

**Hátrány:** Előfordulhat, hogy több táblában kell keresni (például ha a tantermek nevére és férőhelyére vagyunk kíváncsiak, akkor össze kell kapcsolni a táblákat).

# Objektumorientált megközelítés --- 2

1. Minden altípushoz külön tábla felvétele, egy egyed csak egy táblában szerepel. Az altípusok öröklík a főtípus attribútumait.

(Objektumorientált stílusú reprezentálás)

HELYISÉG (épület, ajtószám, név, alapterület)

TANTEREM (épület, ajtószám, név, alapterület, férőhely, tábla, vetítő)

GÉPTEREM (épület, ajtószám, név, alapterület, gépszám)

IRODA (épület, ajtószám, név, alapterület, telefon, fax)

DOLGOZÓ (adószám, név, lakcím, *épület*, *ajtószám*)

Hátrányok:

- Kereséskor gyakran több táblát kell vizsgálni (ha például a D épület 803. számú terem alapterületét keressük).
- Kombinált altípus (például számítógépes tanterem) csak új altípus felvételével kezelhető.

# Nullértékek használata

## relációk egyesítéséhez --- 3

3. Egy közös tábla felvétele, az attribútumok uniójával.  
Az aktuálisan értékkel nem rendelkező attribútumok **NULL** értékűek.

(Reprezentálás nullértékekkel)

HELYISÉG (épület, ajtószám, név, alapterület, férőhely, tábla,  
vetítő, gépszám, telefon, fax)

DOLGOZÓ (adószám, név, lakcím, *épület*, *ajtószám*)

Hátrányok:

- Az ilyen egyesített táblában általában sok NULL attribútumérték szerepel.
- Elveszíthetjük a típusinformációt (például ha a gépteremnél a gépszám nem ismert és ezért NULL, akkor a gépterem lényegében az egyéb helyiségek kategóriájába kerül).

# Kérdés/Válasz

- Köszönöm a figyelmet! Kérdés/Válasz?
- Házi feladat: Gyakorlás az 8. és 9. előadások két hetére:
- Ezen a héten: **DML-utasítások, tranzakciók** (lásd 5EA)
  - Változóhasználat (Példatár 4.fej., + PL/SQL: 8.fej. is)
  - DML-utasítások: insert, update, delete (Példatár 5.fej.)
  - Adatbázis-tranzakciók: commit, rollback, savepoint
- Következő héten: **DDL-utasítások** (lásd a köv. 9EA)
  - DDL-utasítások: adattáblák létrehozása, módosítása, integritási megszorítások (Példatár 5.fejezet folyt.) és
  - Nézet-tábla létrehozása és törlése, táblák tartalmának módosítása nézet-táblákon keresztül (Példatár 6.fej.)

<http://people.inf.elte.hu/sila/eduAB/Feladatok.pdf>