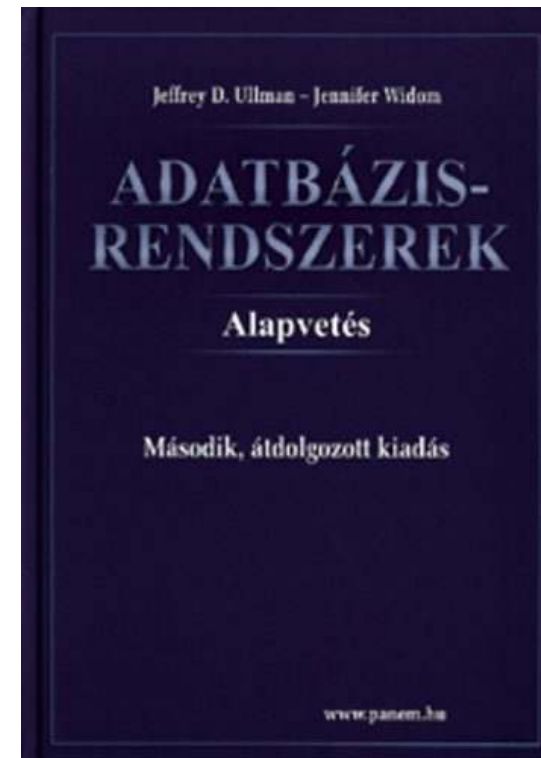


Az egyed-kapcsolat modell (E/K)

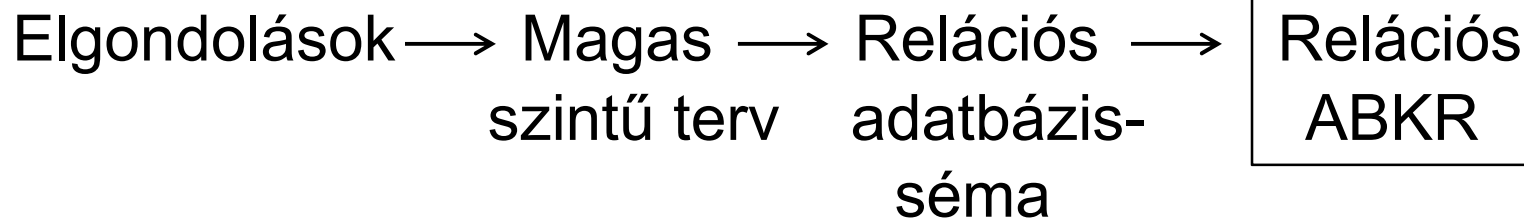
Tankönyv: Ullman-Widom:
Adatbázisrendszerek Alapvetés
Második, átdolgozott kiadás,
Panem, 2009

- 4.1. Az egyed-kapcsolat (E/K) modell
- 4.2. Tervezési alapelvek
- 4.3. Megszorítások modellezése
- 4.4. Gyenge egyedhalmazok



Magas szintű adatbázismodellek

- 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?

Magas szintű adatbázismodellek

- Tervezési fázis
 - A lehetőségeket értékelni, elképzeléseket egyeztetni
- **A 3 szintű ANSI/SPARC architektúra**
 - **Logikai** (külső, a felhasználói szemléletnek megfelel)
 - **Fogalmi** (absztrakt, szintetizálja az összes felhasználói szemléletet)
 - **Fizikai** (belső, az adatbázis valamilyen fizikai adatstruktúrában letárolva a háttértárolón)

Az adatbázis-sématervezés lépései

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

- ❑ a feldolgozandó információ elemzése,
- ❑ az információk közti kapcsolatok meghatározása
- ❑ az eredmény ábrázolása (E/K diagram),
- ❑ adatbázisterv készítése (transzformációs lépés),
- ❑ adatbázisterv finomítása (összevonások),
- ❑ megszorítások modellezése,
függőségek meghatározása,
- ❑ optimális adatbázisterv készítése,
dekomponálás, normalizálás,
- ❑ az adatbázisterv megvalósítása

Egyed-kapcsolat modell elemei

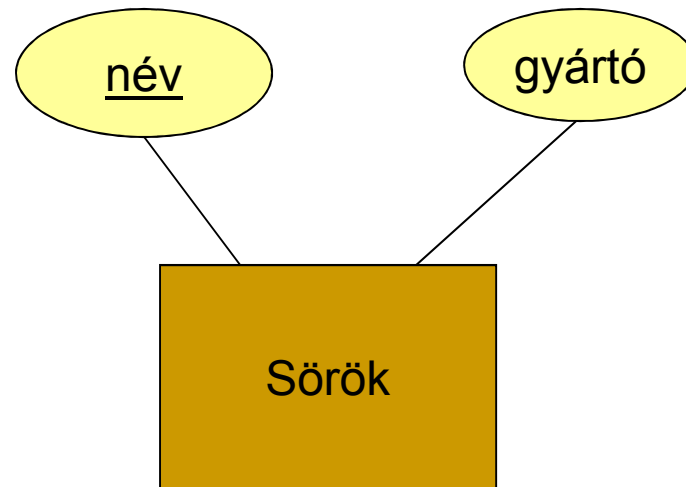
- **Egyed-kapcsolat modell E/K modell**
(Entity-relationship ER) alapfogalmak:
- **Egyedhalmazok**
 - Miről gyűjtünk adatokat?
 - Mit tegyünk egy gyűjteménybe? - hasonlóság
 - Hasonló egyedek összessége
- **Attribútumok**
 - Megfigyelhető tulajdonságok, megfigyelt értékek
 - Az egyedek tulajdonságait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

Egyed-kapcsolat modell elemei

- $E(A_1, \dots, A_n)$ egyedhalmaz **séma**,
 - E az egyedhalmaz neve,
 - A_1, \dots, A_n tulajdonságok,
 - $DOM(A_i)$ – 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
 - $e_i(k) \in DOM(A_k)$,
 - semelyik két egyed nem egyezik meg minden attribútumban (az összes tulajdonság szuperkulcs),
minimális szuperkulcs = kulcs

Egyed-kapcsolat (E/K) diagram

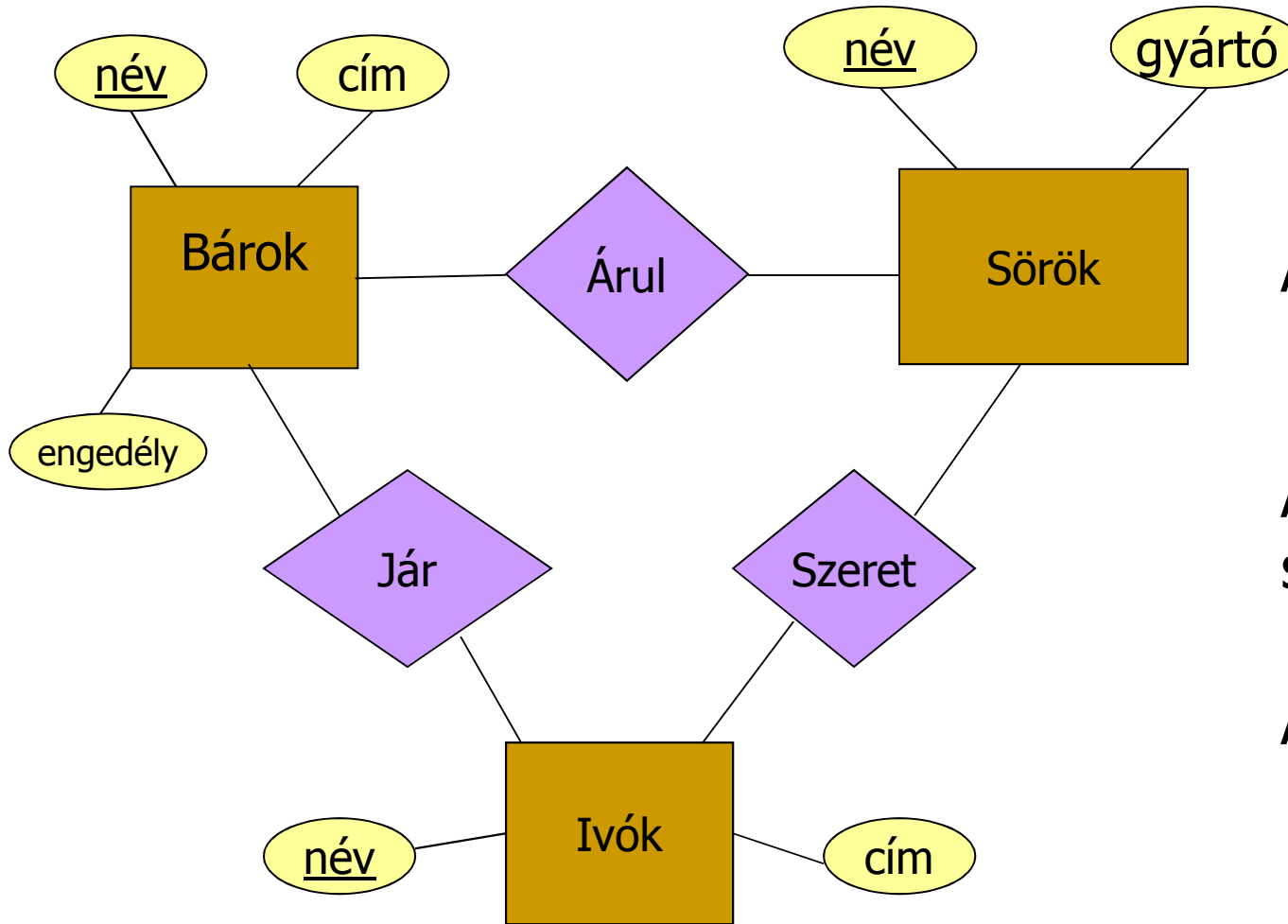
- Egyedhalmazok, kapcsolatok, típusok, egyéb feltételezések ábrázolása.
- **egyedhalmaz**
 - az elsődleges kulcshoz tartozó tulajdonságokat aláhúzzuk.



Egyed-kapcsolat modell elemei

- $K(E_1, \dots, E_p)$ 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.

Példa: Kapcsolatok



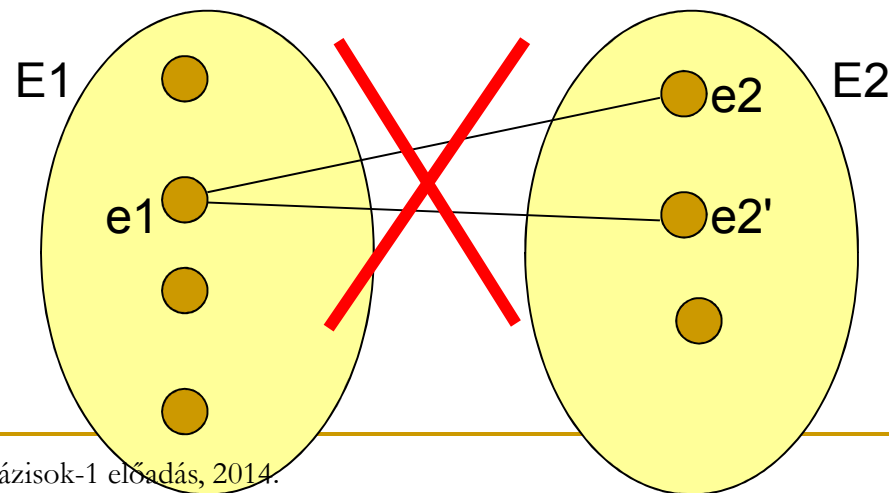
A bárok söröket árulnak.

Az ivók söröket szeretnek.

Az ivók bárokba járnak.

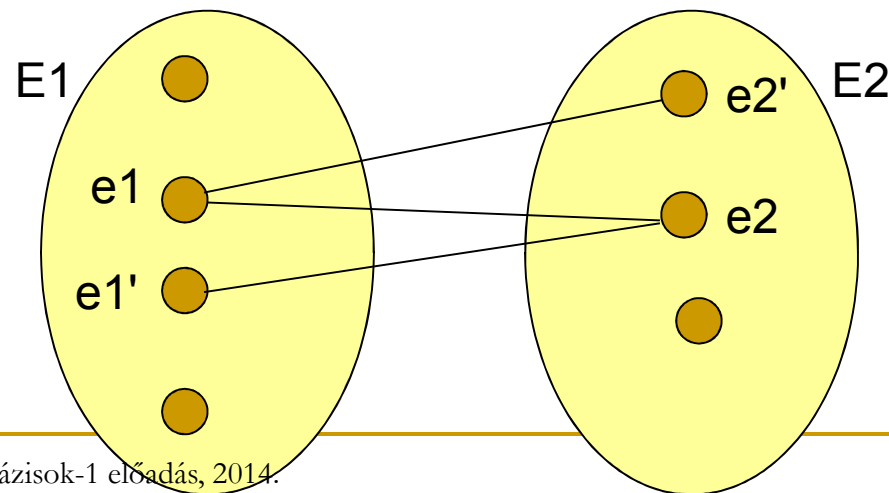
Kapcsolatok típusai

- $K(E1, E2)$ bináris kapcsolat, sok-egy (n:1)
 - ❑ $K \{(e_i, e_j)\}$ alakú előfordulásaiban nem szerepelhet egyszerre (e_1, e_2) és (e_1, e_2') , ha e_2 és e_2' különböznek,
 - ❑ másképpen: **K előfordulásaiban minden $E1$ -beli egyedhez legfeljebb 1 $E2$ -beli egyed tartozhat,**
 - ❑ például: született(név, ország).



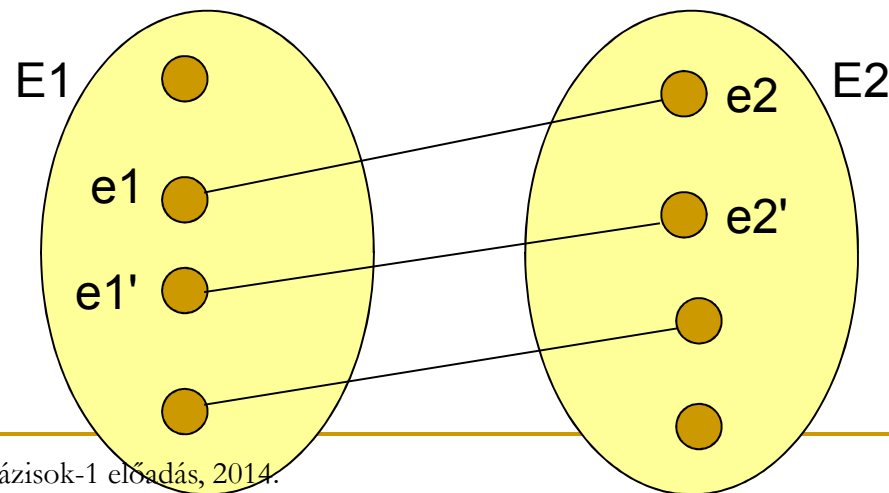
Kapcsolatok típusai

- $K(E1, E2)$ 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).

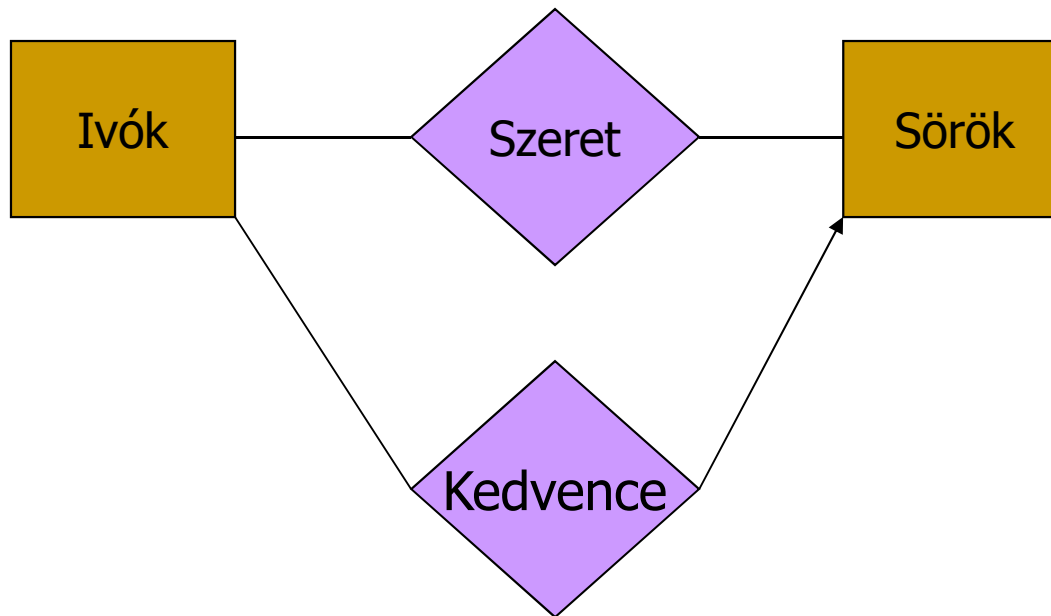


Kapcsolatok típusai

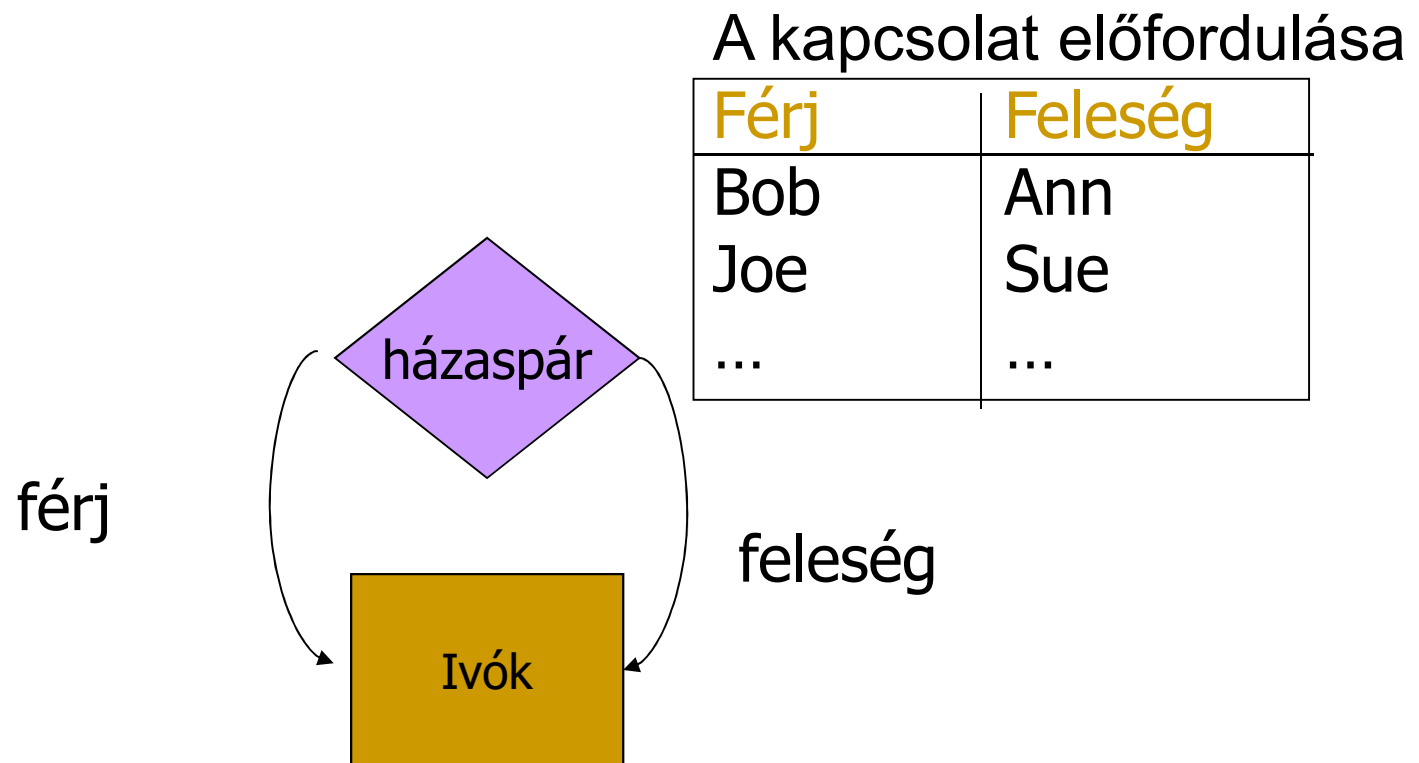
- $K(E1, E2)$ bináris kapcsolat, **egy-egy** (1:1),
 - $K \{(e_i, e_j)\}$ alakú előfordulásai egyszerre sok-egy és egy sok típusúak, vagyis **minden E1-beli egyedhez legfeljebb egy E2-beli egyed tartozhat, és fordítva, minden E2-beli egyedhez legfeljebb egy E1-beli egyed tartozhat,**
 - nem kötelezően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



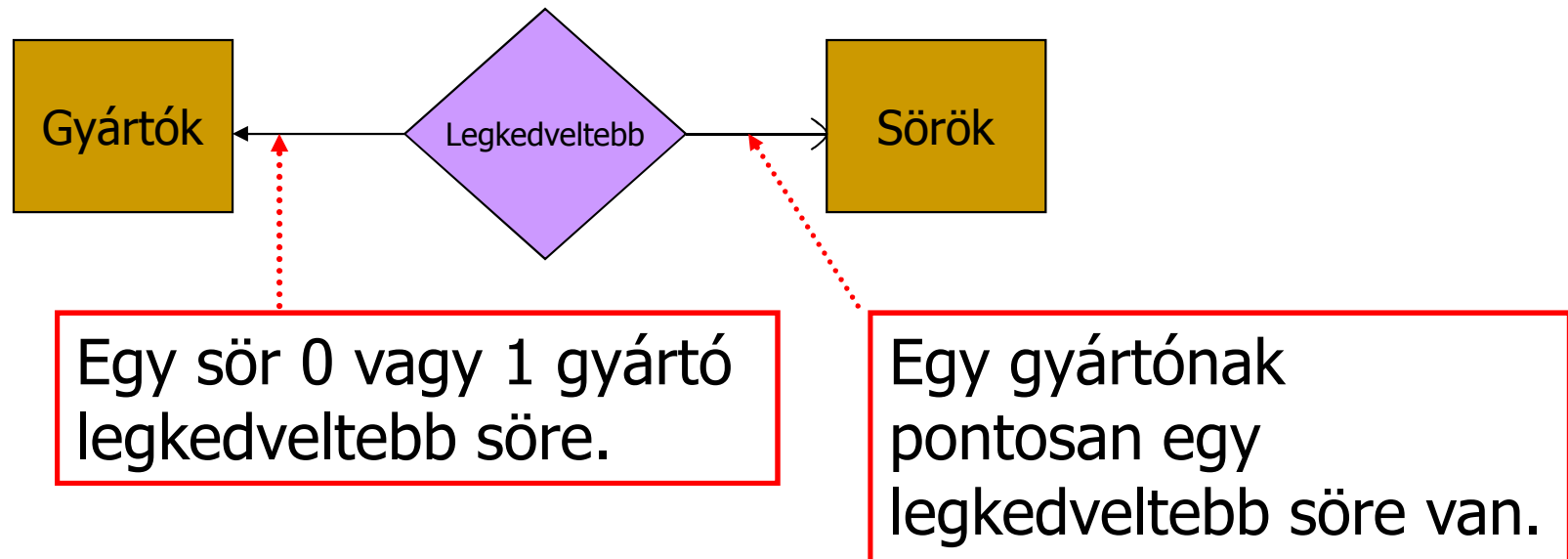
Példa: Két egyedhalmaz között több kapcsolat is lehet



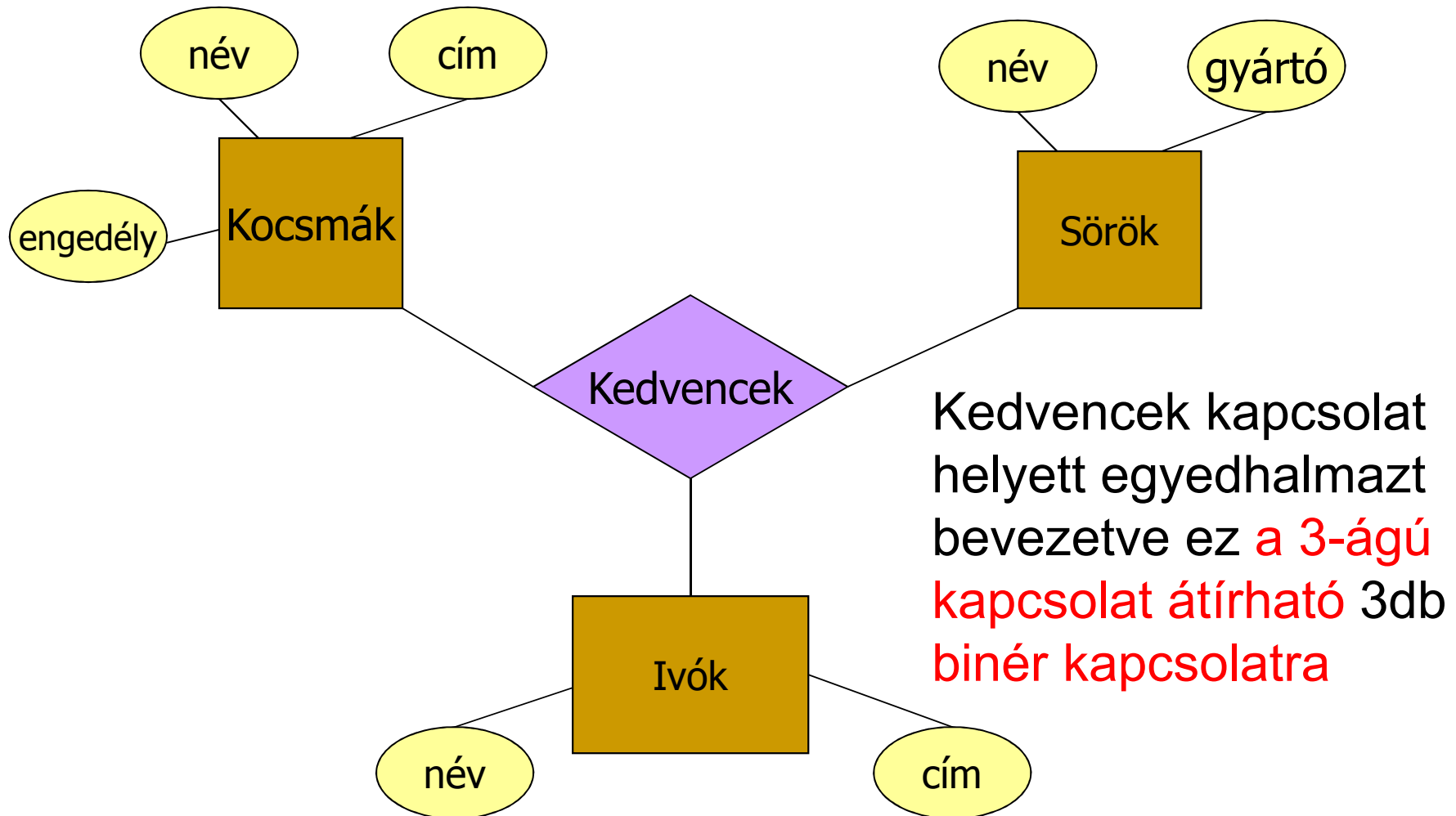
Példa: Egy egyedhalmaz önmagával is kapcsolódhat: Szerepek (Roles)



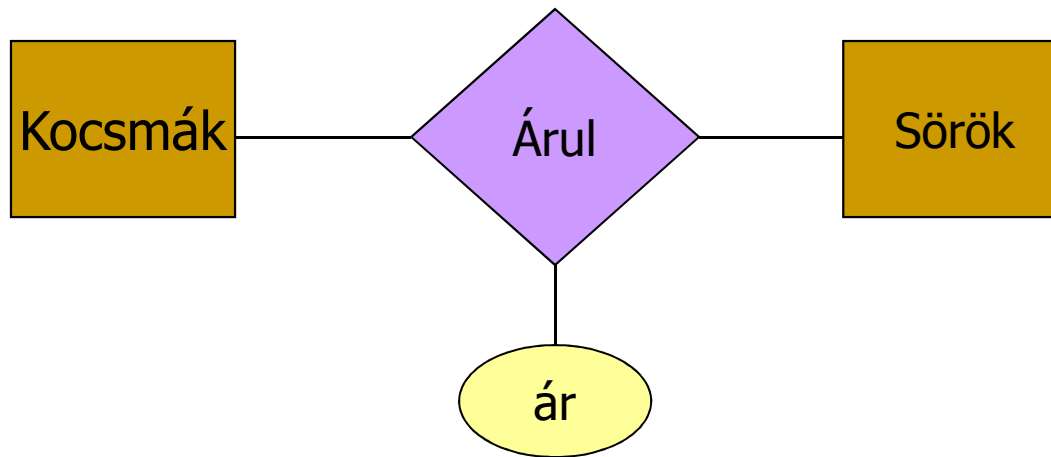
Példa: hivatkozási épség megszorítás jele a kerek végződés —)



Példa: 3-ágú kapcsolatra

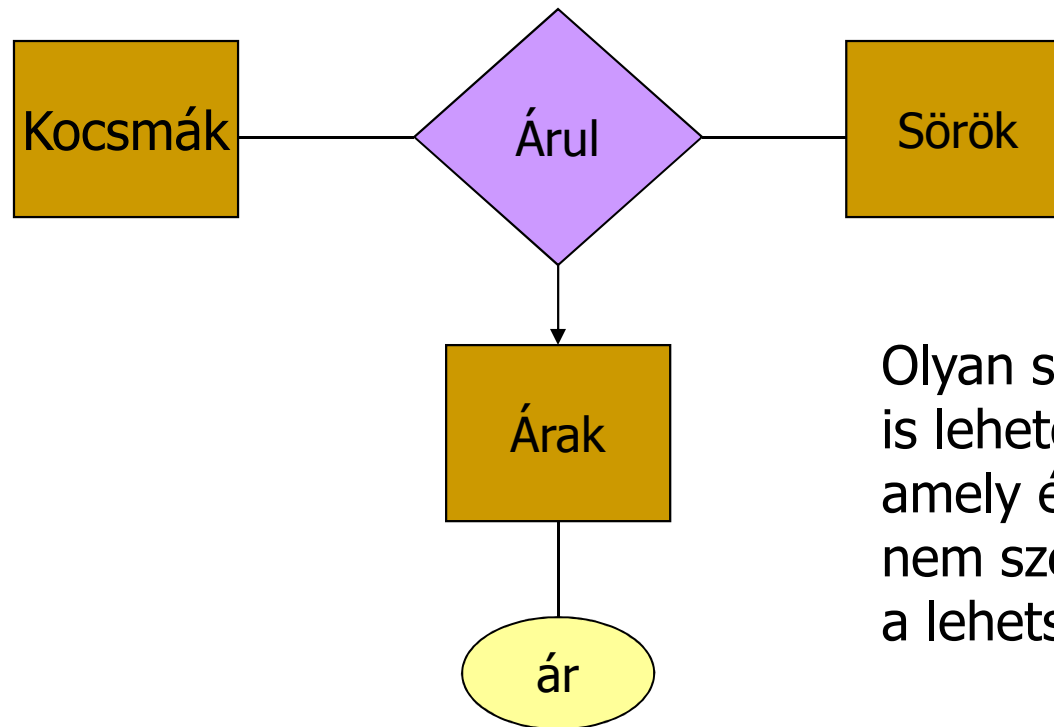


Példa: Kapcsolatnak is lehet attribútuma



Az **ár** a **kocsmák** és **sörök** **együttes** függvénye, de egyiké sem külön.

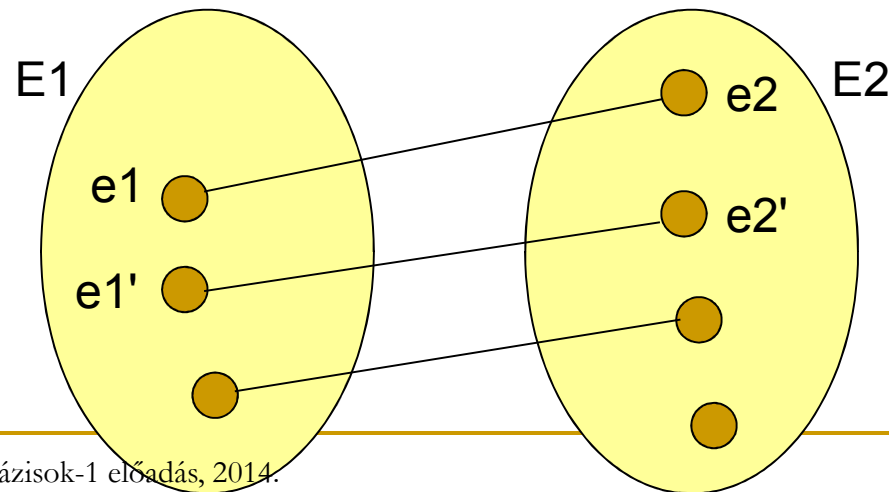
Tervezés: Attribútum vagy egyedosztály?



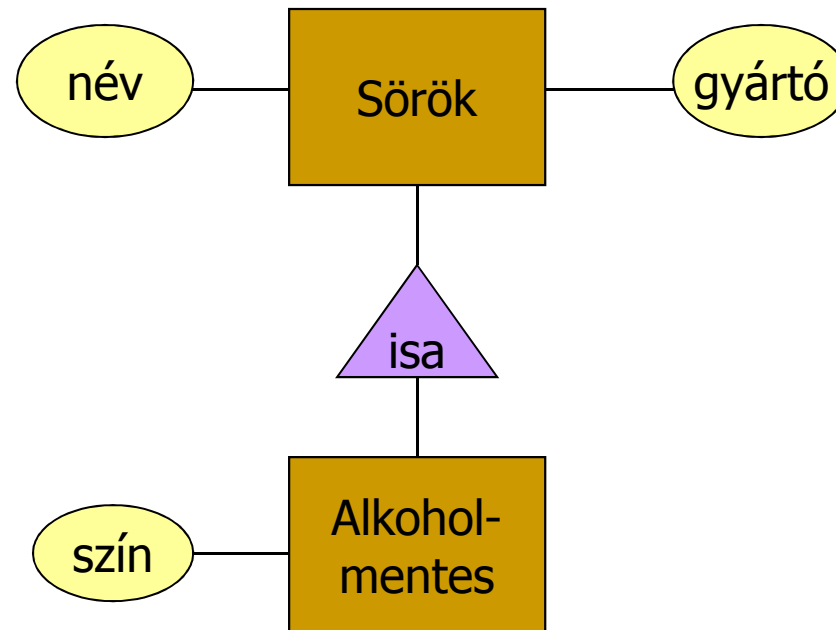
Olyan számokat (árakat) is lehetőségünk van tárolni, amely értékek még árként nem szerepelnek, de csak ezek a lehetséges árértékek.

ISA (az-egy) kapcsolat

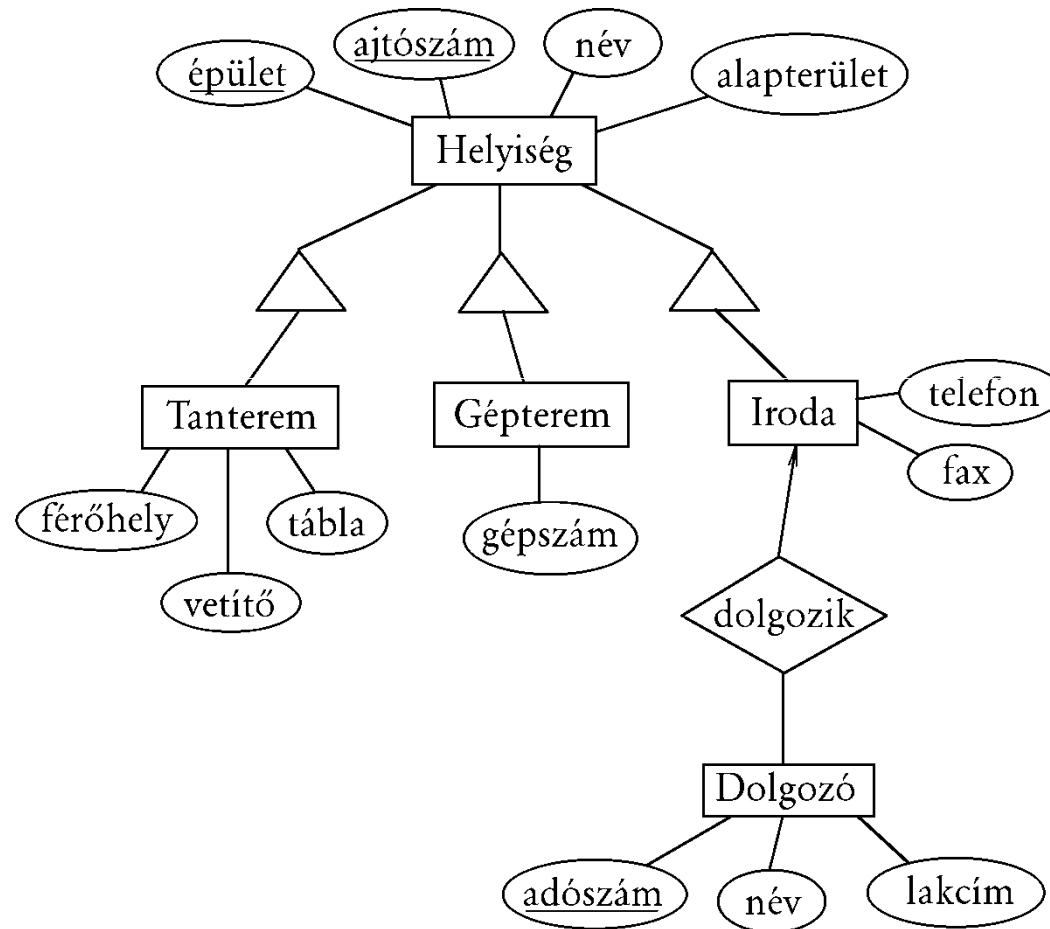
- $K(E1, E2)$ bináris kapcsolat,
 - öröklődési kapcsolat ("az egy", ISA),
 - "a PC is a computer" = "a PC az egy számítógép",
 - speciális egy-egy kapcsolat,
 - $K \{(e_i, e_j)\}$ alakú előfordulásaiban az összes E1-beli egyed szerepel,
 - például: az_egy(főnök, dolgozó).



Példa: Alosztályok és öröklődés

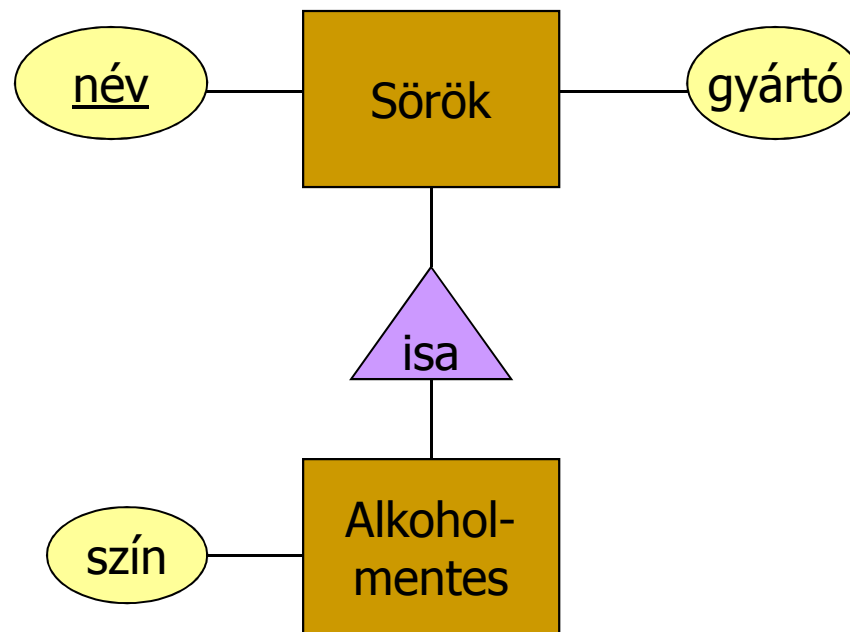


Példa: "Az egy" isa kapcsolatra

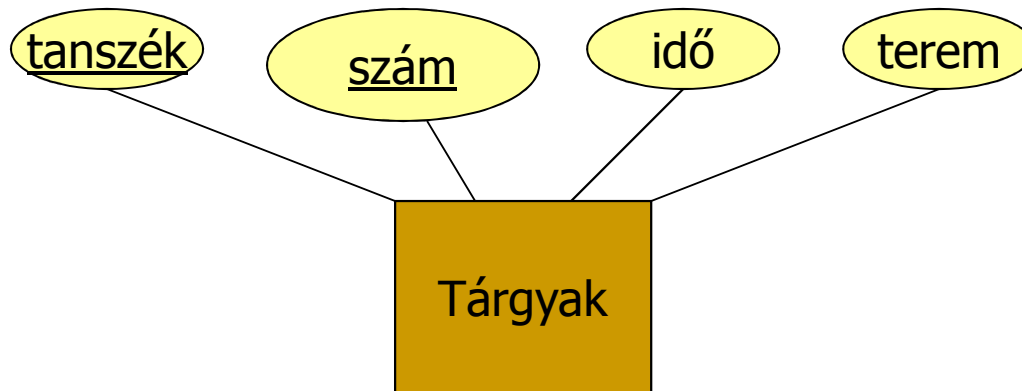


Példa: Kulcsok

A név kulcsa a Sörök-nek

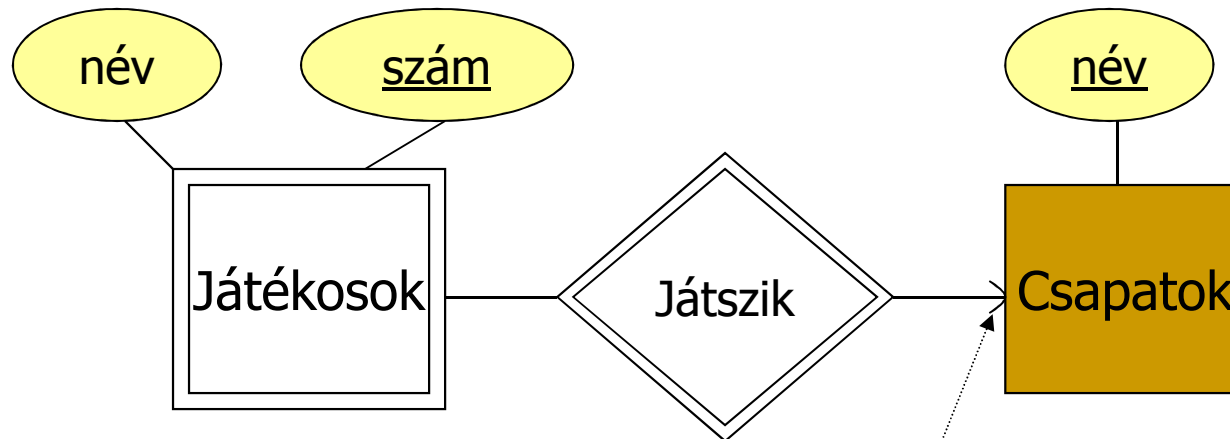


Példa: összetett kulcsok



- Az elsődlegesen kívül lehet más kulcs is:
- (idő, terem) is összetett kulcs.

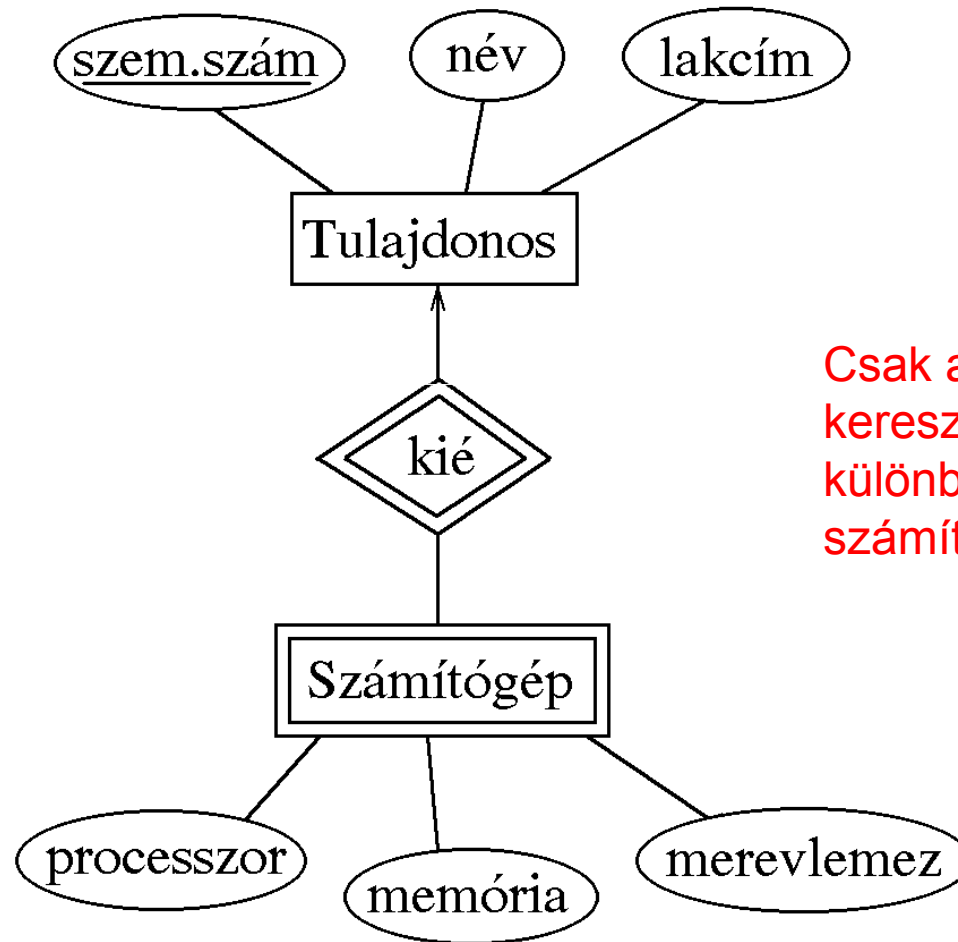
Erős és gyenge egyedhalmaz



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

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

Gyenge egyedhalmazok



Csak a "kié" kapcsolaton keresztül különböztethetők meg a számítógépek.

Általánosítás

- A kapcsolatnak is lehetnek saját tulajdonságai
 - $K(E_1, \dots, E_p, A_1, \dots, A_q)$, ahol A_1, \dots, A_q tulajdonságok,
 - például: $\text{index}(\text{diák}, \text{tárgy}, \text{jegy}, \text{dátum})$, ahol a jegy és dátum saját tulajdonság.
- **Általánosított egy-sok többágú kapcsolat**
 - $K(E_1, \dots, E_p)$ kapcsolat előfordulásaiban nem szerepelhet egyszerre (e_1, e_2, \dots, e_p) és (e_1', e_2, \dots, e_p) , ha e_1 és e_1' különböznek, vagyis **minden (E_2, \dots, E_p) -beli $p-1$ -eshez csak 1 E_1 -beli egyed tartozhat**,
 - hasonlóan értelmezhető bármelyik $p-1$ egyedosztályra,
 - például: $\text{szállít}(\text{szállító}, \text{áru}, \text{ár})$, ahol feltesszük, hogy egy szállító egy adott árut csak egyféle áron szállíthat

Szuperkulcsok, kulcsok, azonosítók

- Az egyedhalmaz **szuperkulcsa egy azonosító**, vagyis olyan tulajdonság-halmaz, amelyről feltehető, hogy az egyedhalmaz előfordulásaiban nem szerepel két különböző egyed, amelyek ezeken a tulajdonságokon megegyeznek.
- Az összes tulajdonság mindig szuperkulcs.
- **A minimális szuperkulcsot kulcsnak nevezzük.**
- Az egyedhalmaz (szuper)kulcsai azonosításra használhatók.
- Több (szuper)kulcs is lehet. Ezek közül egyet kiválasztunk **elsődleges kulcsnak**, a többi **másodlagos (szuper)kulcs**.
- Ha E_1 egyedosztályban szereplő T_1, \dots, T_k tulajdonságok halmaza az E_2 egyedosztály kulcsa, akkor azt mondjuk, hogy **T_1, \dots, T_k idegen kulcsa az E_1 -nek az E_2 -re nézve**,
 - például: diák(etr_kód,név,anya_neve,szül_idő,lakcím) egyedosztályban a **név,anya_neve,szül_idő** idegen kulcs az ösztöndíjas(név,anya_neve,szül_idő,év,ösztöndíj) egyedosztályra nézve.

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.