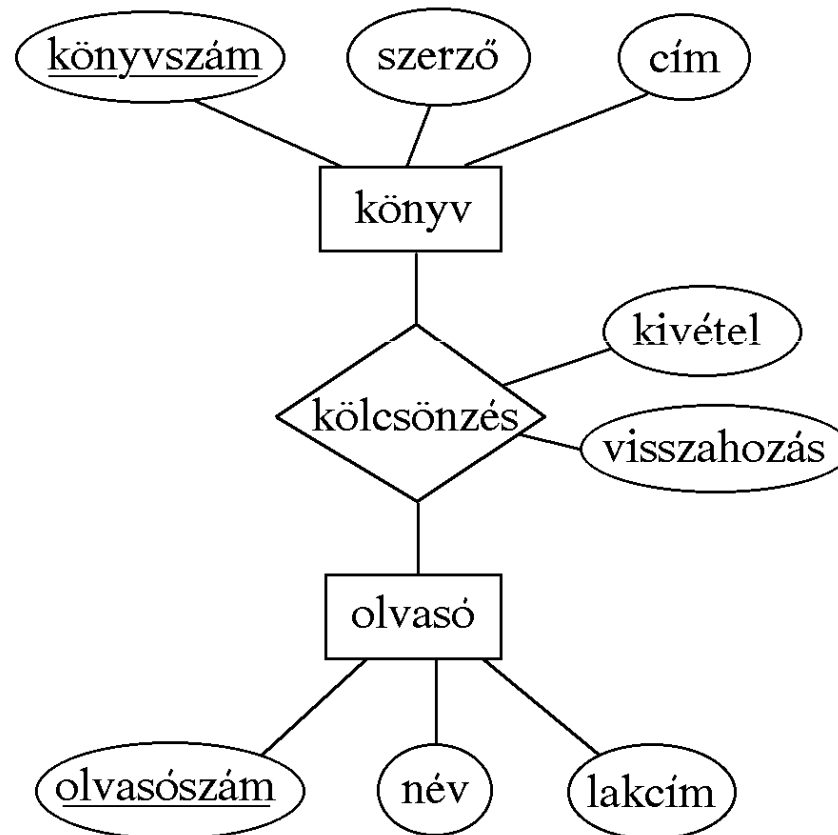


# Adatmodellezés, adatbázis-tervezés

## Az adatbázis-tervezés lépései:

1. a feldolgozandó **információ elemzése**,
2. az információk közti **kapcsolatok meghatározása**,
3. az eredmény ábrázolása (**E/K diagram**),
4. **adatbázisterv készítése** (transzformációs lépés),
5. adatbázisterv finomítása (**összevonások**),
6. megszorítások modellezése, **függőségek meghatározása**,
7. optimális adatbázisterv készítése (**dekomponálás, normalizálás**),
8. az **adatbázisterv megvalósítása** SQL-ben  
(create table..., create view ..., stb.).

# Egy könyvtár adatmodellje



# Egyed-kapcsolat formális modell

- **$E(A_1, \dots, A_n)$  egyedosztály **séma**,**

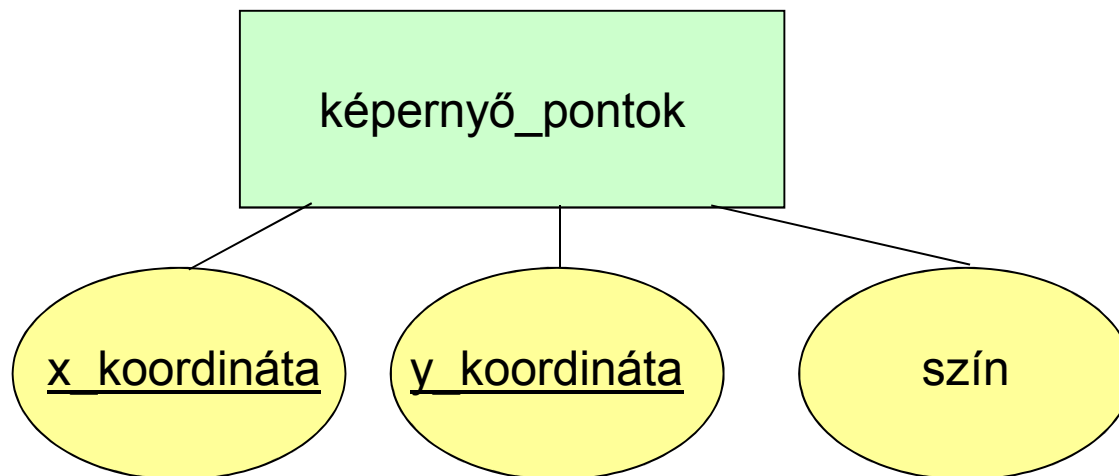
  - E az egyedosztály neve,
  - $A_1, \dots, A_n$  tulajdonságok,
  - $DOM(A_i)$  – lehetséges értékek halmaza.
  - például: tanár(név, tanszék).

- **$E(A_1, \dots, A_n)$  sémájú egyedosztály **előfordulása**:**

  - $E = \{e_1, \dots, e_m\}$  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

- Egyedosztályok, kapcsolatok, típusok, egyéb feltételezések ábrázolása.
- **egyedosztály**
  - az elsődleges (szuper)kulcshoz tartozó tulajdonságokat aláhúzzuk.



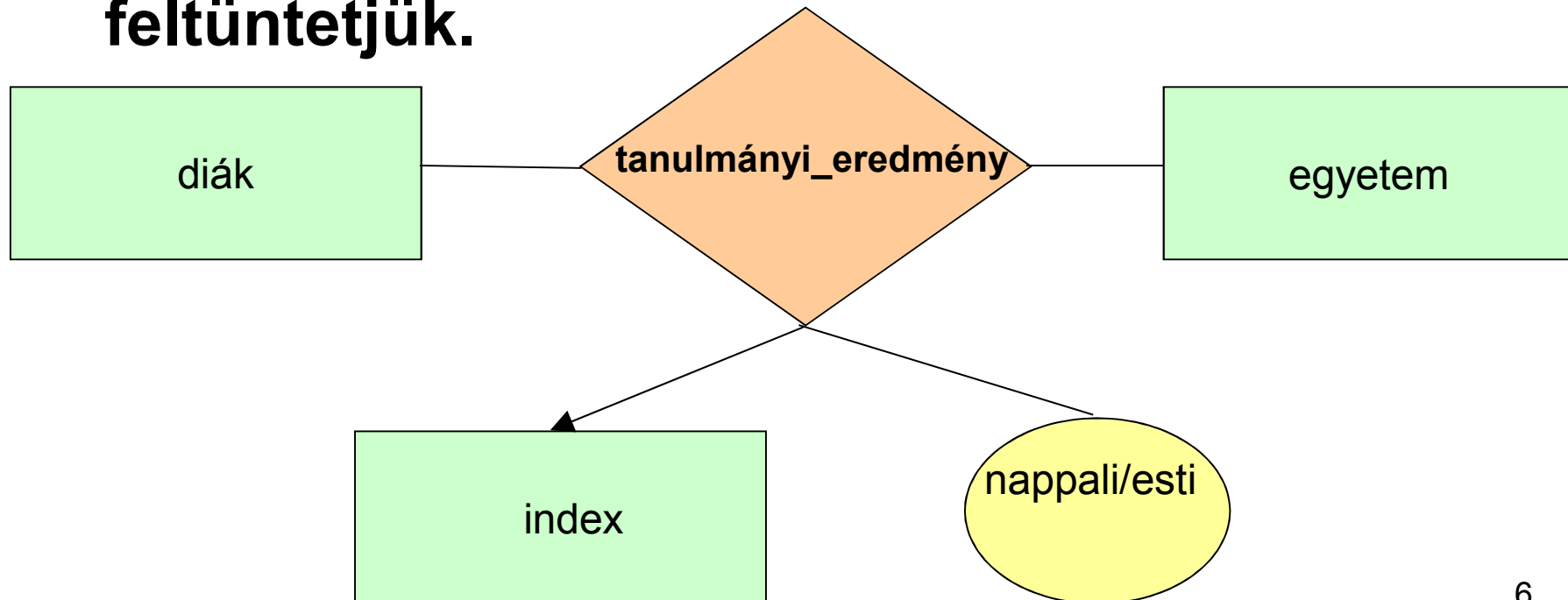
# Egyed-kapcsolat formális modell

- **$K(E_1, \dots, E_p)$  a kapcsolat sémája,**
  - $K$  a kapcsolat neve,
  - $E_1, \dots, E_p$  egyedosztályok 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.

# Egyed-kapcsolat (E/K) diagram

- **kapcsolat**

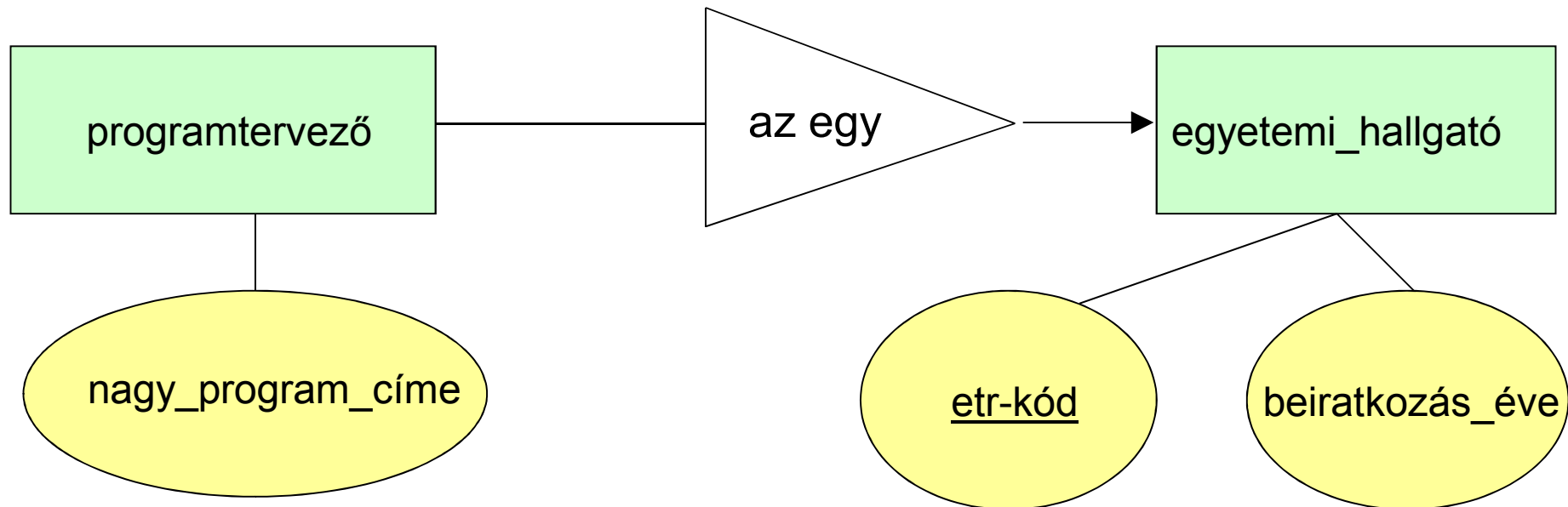
- a típusban szereplő "egy" végponthoz nyilat húzunk, a kapcsolat saját tulajdonságait is feltüntetjük.



# Egyed-kapcsolat (E/K) diagram

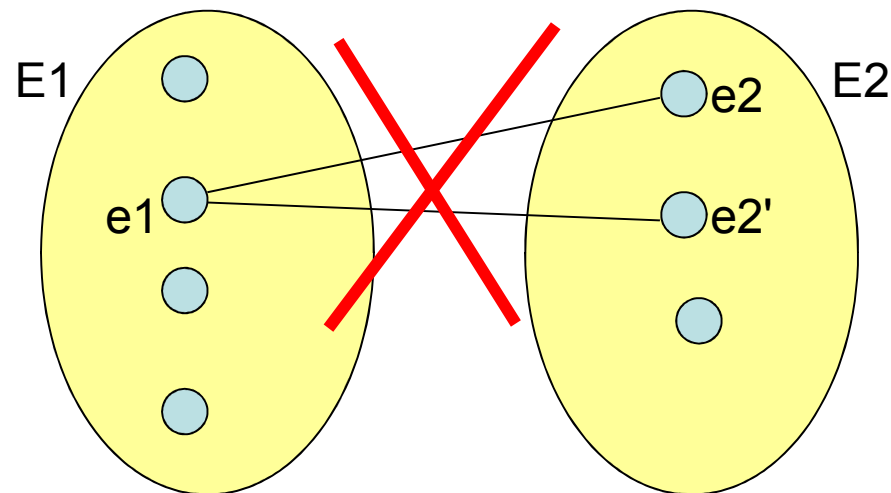
- **kapcsolat**

- öröklődési típus esetén az általánosabb egyedosztály felé húzunk nyilat, és a speciális osztálynak (alosztálynak) csak az általánostól különböző tulajdonságait adjuk meg.



# Kapcsolatok típusai

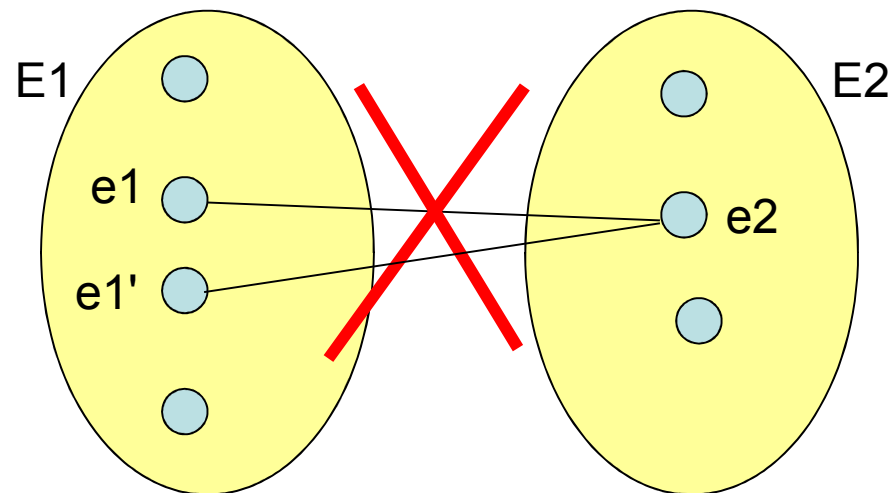
- **$K(E1, E2)$  bináris kapcsolat,**
  - sok-egy (n:1)
    - $K \{(e_i, e_j)\}$  alakú előfordulásaiban nem szerepelhet egyszerre  $(e_1, e_2)$  és  $(e_1, e_2')$ , ha  $e_2$  és  $e_2'$  különböznek,
    - másképpen: **K előfordulásaiban minden E1-beli egyedhez legfeljebb 1 E2-beli egyed tartozhat,**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





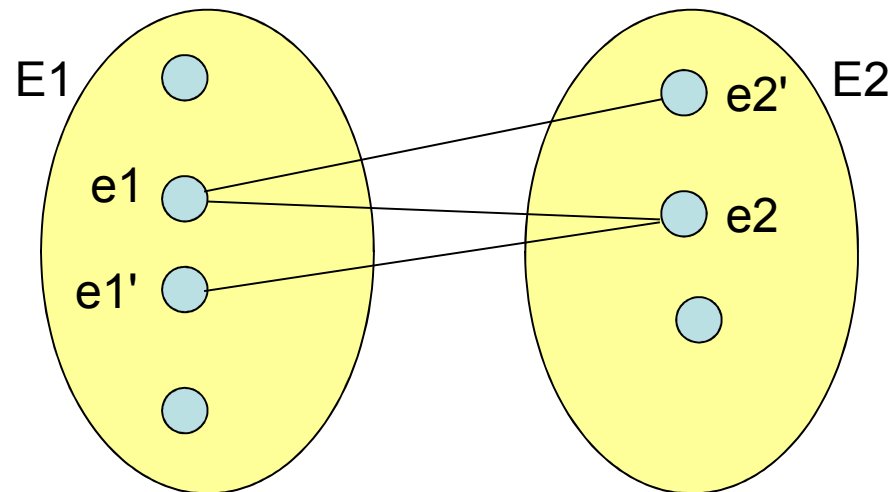
# Kapcsolatok típusai

- **$K(E1, E2)$  bináris kapcsolat,**
  - **egy-sok** (1:n) (vagy (1:m)),
    - $K \{(e_i, e_j)\}$  alakú előfordulásaiban nem szerepelhet egyszerre  $(e_1, e_2)$  és  $(e_1', e_2)$ , ha  $e_1$  és  $e_1'$  különböznek,
    - másképpen:  **$K$  előfordulásaiban minden  $E_2$ -beli egyedhez legfeljebb 1  $E_1$ -beli egyed tartozhat,**
    - például: `vb_gyoztes(ország, rendezo_ország)`.



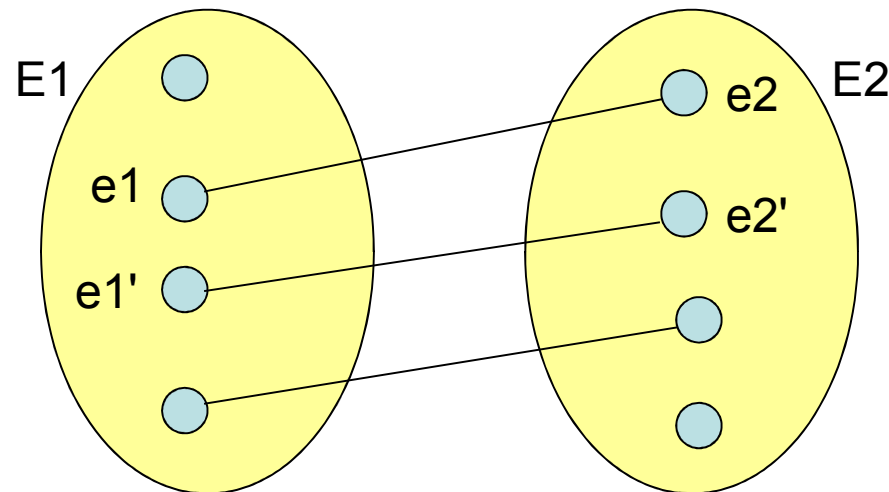
# Kapcsolatok típusai

- **$K(E1, E2)$  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).



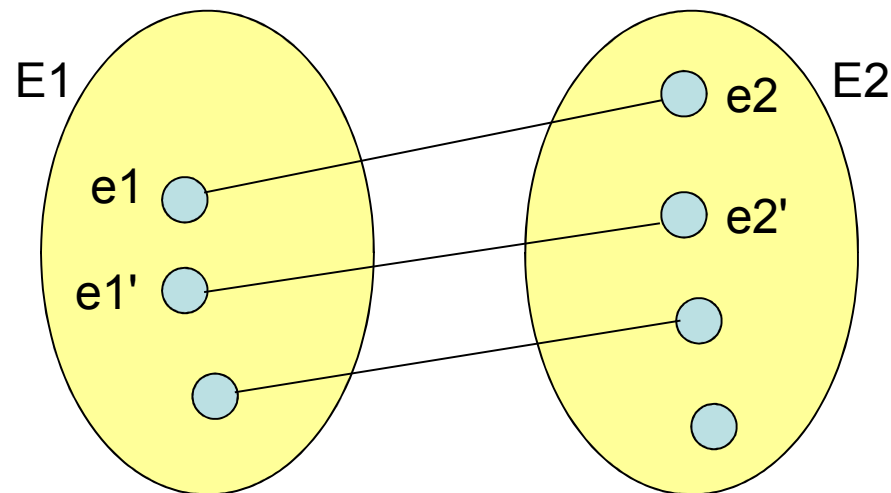
# Kapcsolatok típusai

- $K(E1, E2)$  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).

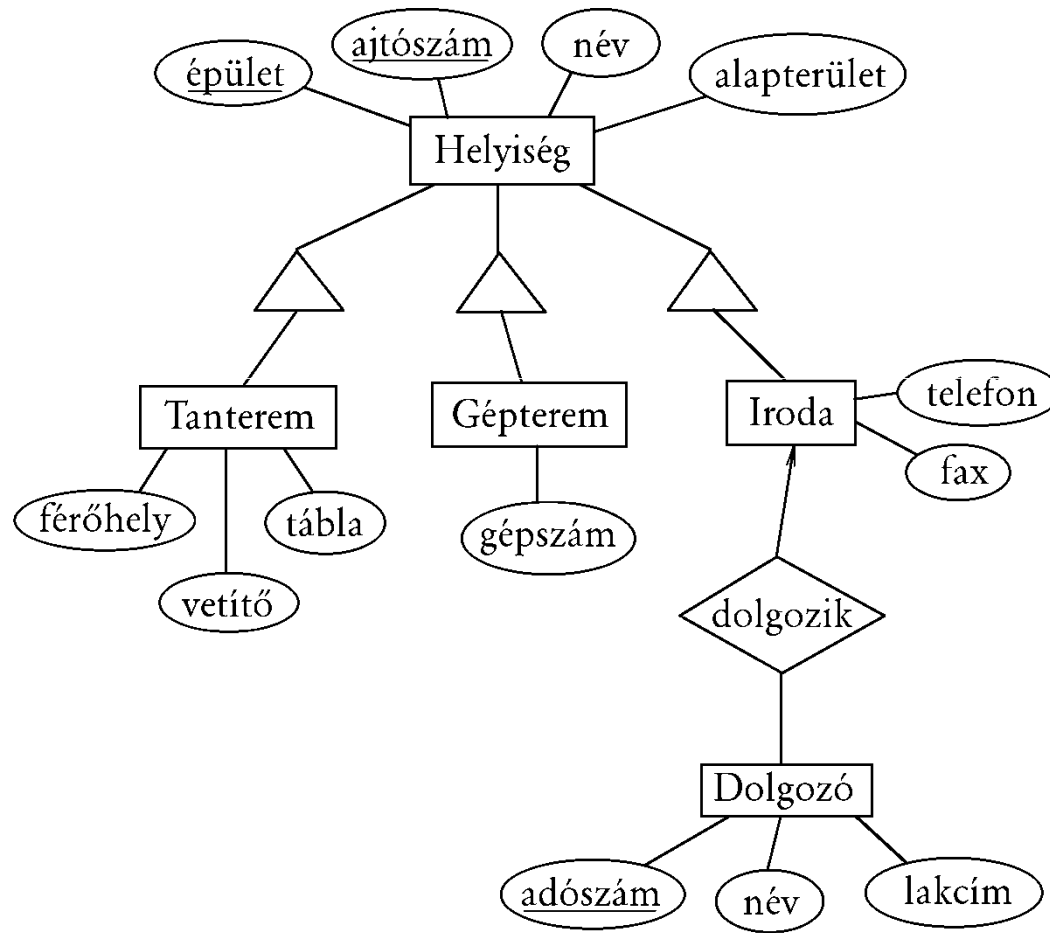


# Kapcsolatok típusai

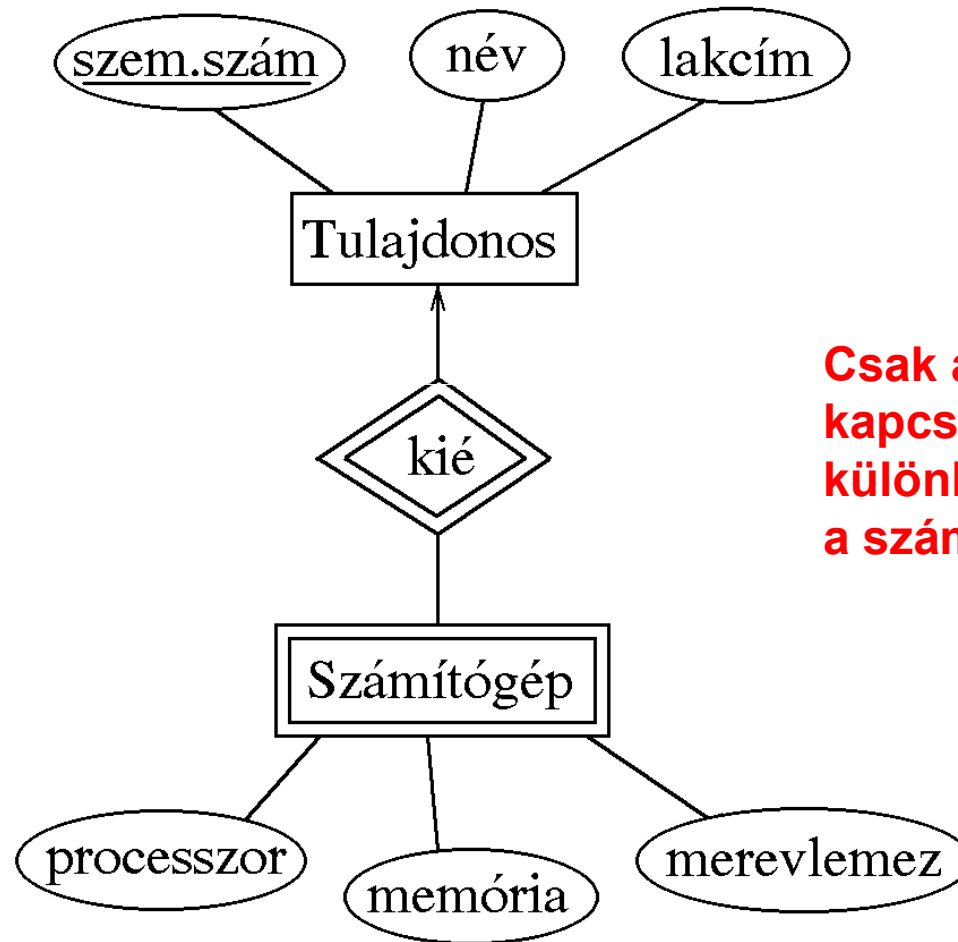
- $K(E1, E2)$  bináris kapcsolat,
  - öröklődési kapcsolat ("az egy", ISA),
  - "a PC is a computer" = "a PC az egy számítógép",
  - **speciális egy-egy kapcsolat,**
  - $K \{(e_i, e_j)\}$  alakú előfordulásaiban **az összes E1-beli egyed szerepel,**
  - például: az\_egy(főnök, dolgozó).



# "Az egy" kapcsolat

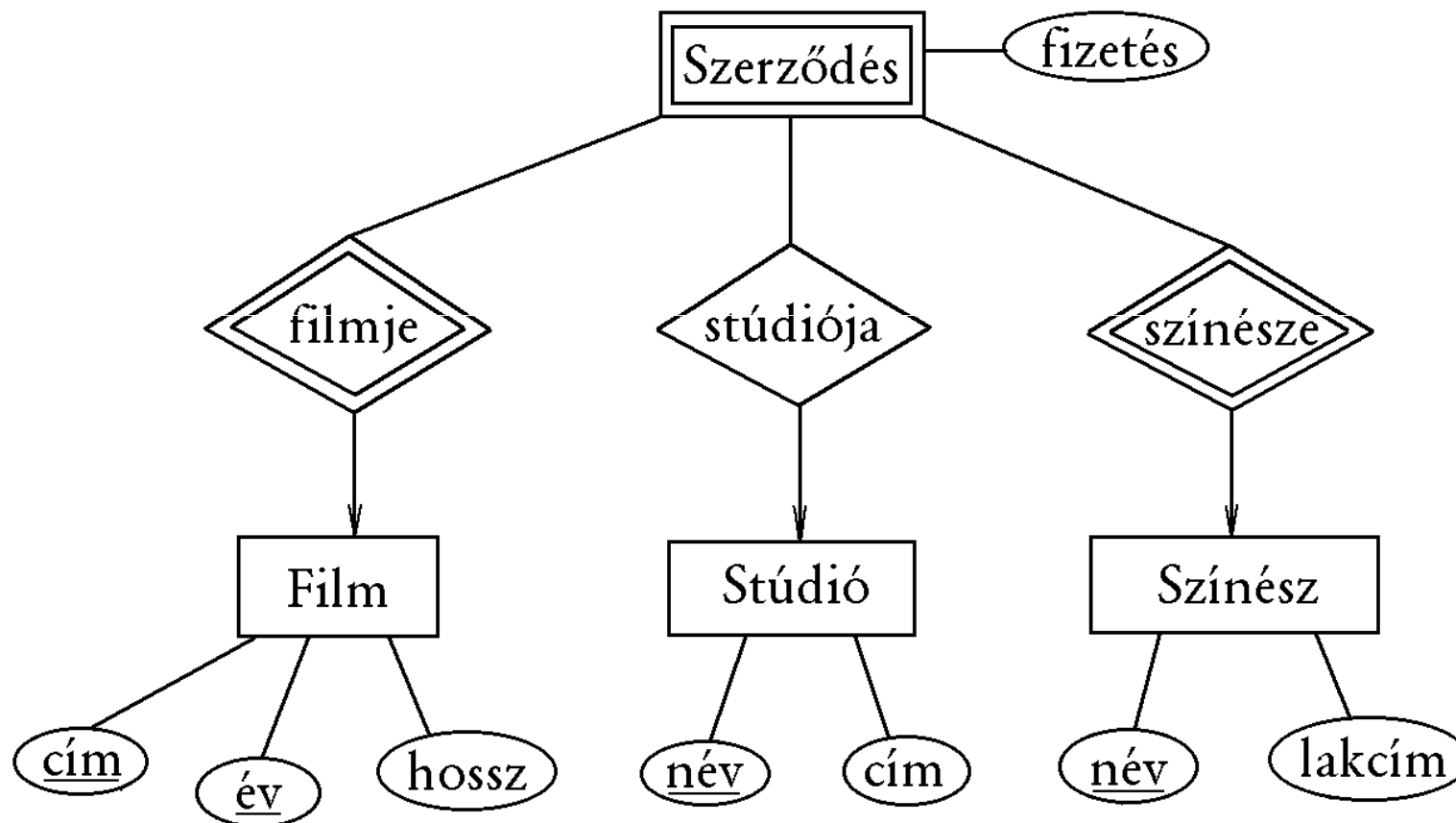


# GYENGE EGYEDOSZTÁLY



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

# Sokágú kapcsolat helyettesítése gyenge egyedosztállyal és bináris kapcsolatokkal



# 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,
    - az ár tekinthető saját tulajdonságnak, vagy  $\text{Ár}(\text{ár})$  egyedosztálynak.



# Szuperkulcsok, kulcsok, azonosítók

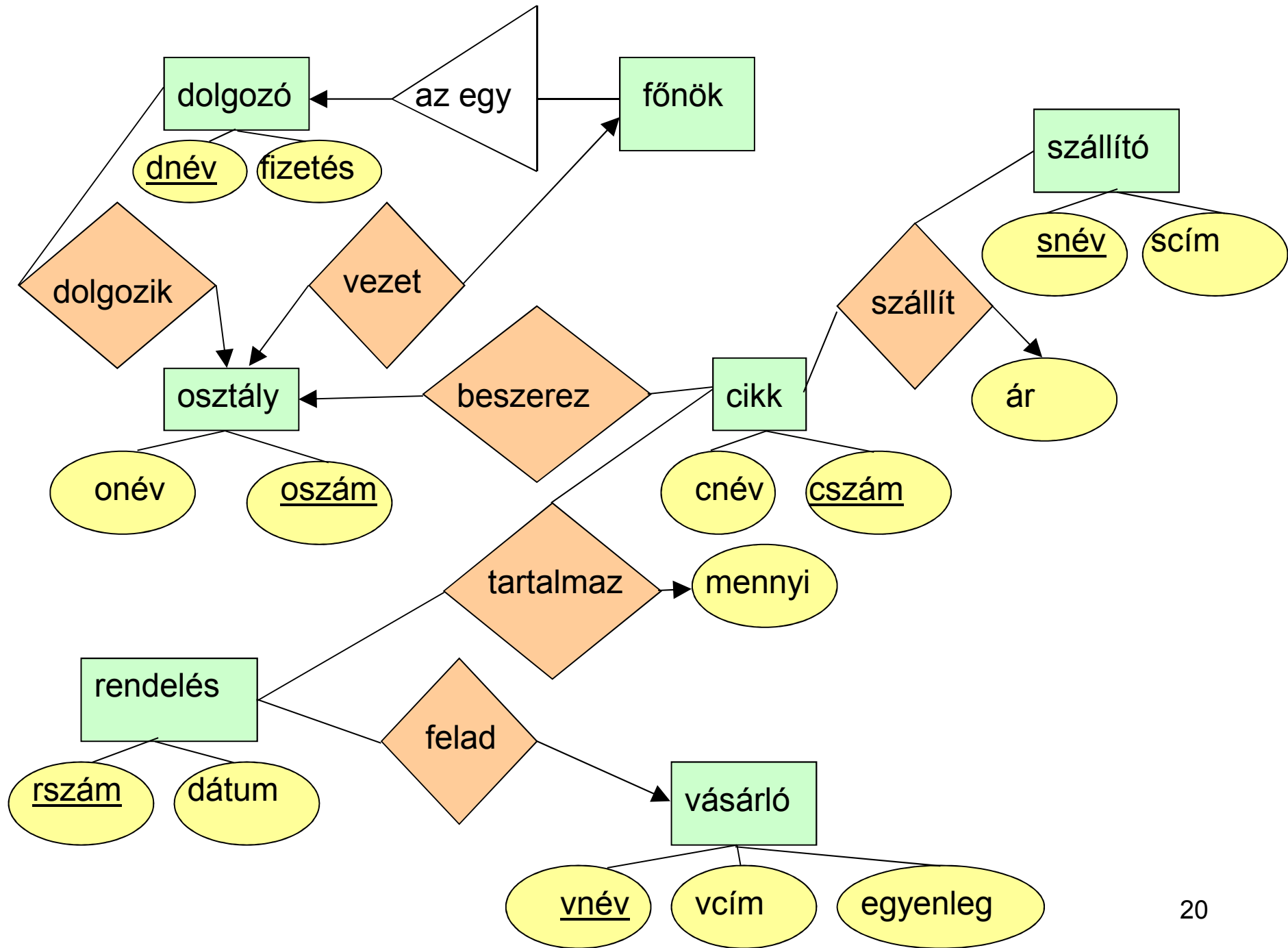
- Az egyedosztály **szuperkulcsa egy azonosító**, vagyis olyan tulajdonság-halmaz, amelyről feltehető, hogy az egyedosztály előfordulásaiban nem szerepel két különböző egyed, amelyek ezeken a tulajdonságokon megegyeznek.
- Az összes tulajdonság mindig szuperkulcs.
- **A minimális szuperkulcsot kulcsnak nevezzük.**
- Az egyedosztály (szuper)kulcsai azonosításra használhatók.
- Több (szuper)kulcs is lehet. Ezek közül egyet kiválasztunk **elsődleges (szuper)kulcsnak**, a többi **másodlagos (szuper)kulcs**.
- Ha E1 egyedosztályban szereplő  $T_1, \dots, T_k$  tulajdonságok halmaza az E2 egyedosztály (szuper)kulcsa, akkor azt mondjuk, hogy  **$T_1, \dots, T_k$  idegen (szuper)kulcsa az E1-nek az E2-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

- **valóságű 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(jegy)` osztály helyett `jegy` tulajdonságot használjunk.

# Példa: E/K diagramra

- **Modellezzük egy áruháznak, dolgozóinak, vevőinek és beszállítóinak rendszerét!**
- **Feltételezések:**
  - az áruház minden osztályát legfeljebb egy ember vezeti,
  - minden dolgozó legfeljebb egy osztályon dolgozik,
  - az áruházak osztályai felelősek az áruk beszerzéséért,
  - minden szállító legfeljebb egyféle áron szállít egy árut,
  - egy rendelést legfeljebb egy vevőhöz tartozhat,
  - minden rendelésen egy cikkhez legfeljebb egy rendelt mennyiség tartozhat.



# E/K diagram átalakítása relációs adatbázisra

## Mi minek felel meg:

- **egyedosztály séma**  $\longleftrightarrow$  **relációséma**  
 $E(A_1, \dots, A_n)$   $\longleftrightarrow$   $E(A_1, \dots, A_n)$
- **tulajdonságok**  $\longleftrightarrow$  **attribútumok**
- **(szuper)kulcs**  $\longleftrightarrow$  **(szuper)kulcs**
- **egyedosztály előfordulása**  $\longleftrightarrow$  **reláció**
- **e egyed**  $\longleftrightarrow$  **( $e(A_1), \dots, e(A_n)$ ) sor**
- **$R(E_1, \dots, E_p, A_1, \dots, A_q)$**   $\longleftrightarrow$   **$R(K_1, \dots, K_p, A_1, \dots, A_q)$**   
**kapcsolati séma**, ahol  $E_i$  egyedosztály,  $A_j$  saját tulajdonság  
**relációséma**, ahol  $K_i$  az  $E_i$  (szuper)kulcsa

**E/K modell**



**Relációs adatmodell**

## E/K diagram átalakítása 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  $R(E_1, E_2)$  sok-egy kapcsolat, akkor  $R(K_1, K_2)$  relációsémának a  $K_1$  szuperkulcsa lesz.
- A **gyenge entitás** relációsémáját bővíteni kell a meghatározó kapcsolat(ok)ban szereplő egyed(ek) kulcsával.

# Az Áruház diagram átalakítása adatbázisútervé

## Az egyedosztályok átalakítása:

- dolgozó(dnév, fizetés)
- főnök(dnév)
- osztály(onév, oszám)
- szállító(snév, scím)
- cikk(cnév, cszám)
- rendelés(rszám, dátum)
- vásárló(vnév, vcím, egyenleg)

## A kapcsolatok átalakítása:

- dolgozik(dnév, oszá
- vezet(dnév, oszá
- beszerez(cszám, oszá
- szállít(cszám, sznév, ár)
- tartalmaz(rszám, cszám, mennyi)
- felad(rszám, vnév)

**Összesen 13  
relációsémát kaptunk!**

# Összevonások

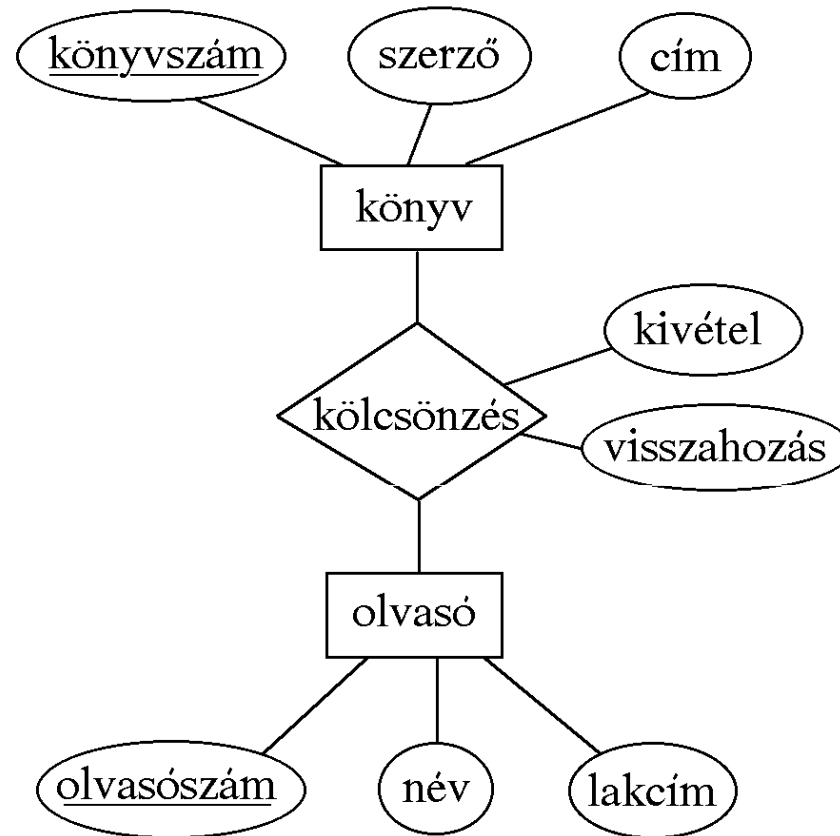
- **Két relációsémát összevonhatunk, ha az egyikben van idegen (szuper)kulcs a másikra nézve.**
- $E1(A1, \dots, An, B1, \dots, Bm)$  és  $E2(B1, \dots, Bm, C1, \dots, Cp)$  helyett  $E3(A1, \dots, An, B1, \dots, Bm, C1, \dots, Cp)$  relációsémát vehetjük, ha  $B1, \dots, Bm$  az  $E2$  elsődleges, vagy másodlagos (szuper)kulcsa.
- Az összevonás eredményét felhasználhatjuk újabb összevonásokban.



# Összevonások eredménye

- dolgozó(dnév,fizetés,oszá
  - osztály(onév,oszá,dnév)
  - szállító(snév,scím)
  - cikk(cnév,cszá,oszá)
  - rendelés(rszá,dátum,vnév)
  - vásárló(vnév,vcím,egyenleg)
  - szállít(snév,cszá,ár)
  - tartalmaz(rszá,cszá,mennyi)
- Összesen 8 relációsémát kaptunk!

# Példa



**KÖNYV** (könyvszám, szerző, cím)

**OLVASÓ** (olvasószám, név, lakcím)

**KÖLCSÖN** (könyvszám, olvasószám, kivétel, visszahozás)

# Összetett attribútumok leképezése

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

# Többértékű attribútumok leképezése

- Kérdés, hogy **többszerzős könyveket** hogyan tartsunk nyilván az adatbázisban.

**1. 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

# Többértékű attribútumok leképezése

- **2. 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.

# Többértékű attribútumok leképezése

**2. Megadás többértékű attribútumként.**

**b) új tábla felvétele.**

**A KÖNYV (könyvszám, szerző, cím)  
sémát az alábbi két sémával  
helyettesítjük:**

**KÖNYV (könyvszám, cím)**

**SZERZŐ (könyvszám, szerző)**

# Többértékű attribútumok leképezése

## *2. Megadás többértékű attribútumként.*

**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)

# Kapcsolatok leképezése

**1. változat:** Ha egy olvasónak egyszerre csak egy könyvet adnak ki, akkor a kölcsönzés **1:1 kapcsolatot** jelent. Ilyenkor a KÖLCSÖN sémában a *könyvszám* és az *olvasószám* egyaránt kulcs. Továbbá, a *visszahozás* attribútumra nincs szükségünk, mivel a könyv visszahozásával a könyv-olvasó kapcsolat megszűnik.

Tehát, a

**KÖLCSÖN** (könyvszám, olvasószám, kivétel)

vagy a

**KÖLCSÖN** (könyvszám, olvasószám, kivétel)

sémát vehetjük fel a kapcsolathoz.

A KÖLCSÖN sémát az azonos kulcsú sémába olvasztva a

**KÖNYV** (könyvszám, szerző, cím, olvasószám, kivétel)

**OLVASÓ** (olvasószám, név, lakcím)

vagy a

**KÖNYV** (könyvszám, szerző, cím)

**OLVASÓ** (olvasószám, név, lakcím, *könyvszám*, kivétel)

adatbázissémákat kapjuk.



# Kapcsolatok leképezése

**2. változat:** Ha egy olvasó több könyvet is kikölcsönözhet, akkor a könyv-olvasó kapcsolat **N:1 típusú**. Ekkor a KÖLCSÖN sémában csak a *könyvszám* lehet kulcs, ezért a KÖLCSÖN sémát csak a KÖNYV sémába olvaszthatjuk:

**KÖNYV** (könyvszám, szerző, cím, olvasószám, kivétel)

**OLVASÓ** (olvasószám, név, lakcím)

# Kapcsolatok leképezése

**3. változat:** Ha az egyes könyvek korábbi kölcsönzéseit is nyilvántartjuk, akkor nem csak egy olvasóhoz tartozhat több könyv, hanem egy könyvhöz is több olvasó (**N:M kapcsolat**), sőt adott olvasó adott könyvet egymás után többször is kikölcsönözhet. Ezért a **KÖLCSÖN** sémában

{könyvszám, kivétel}

vagy

{könyvszám, visszahozás}

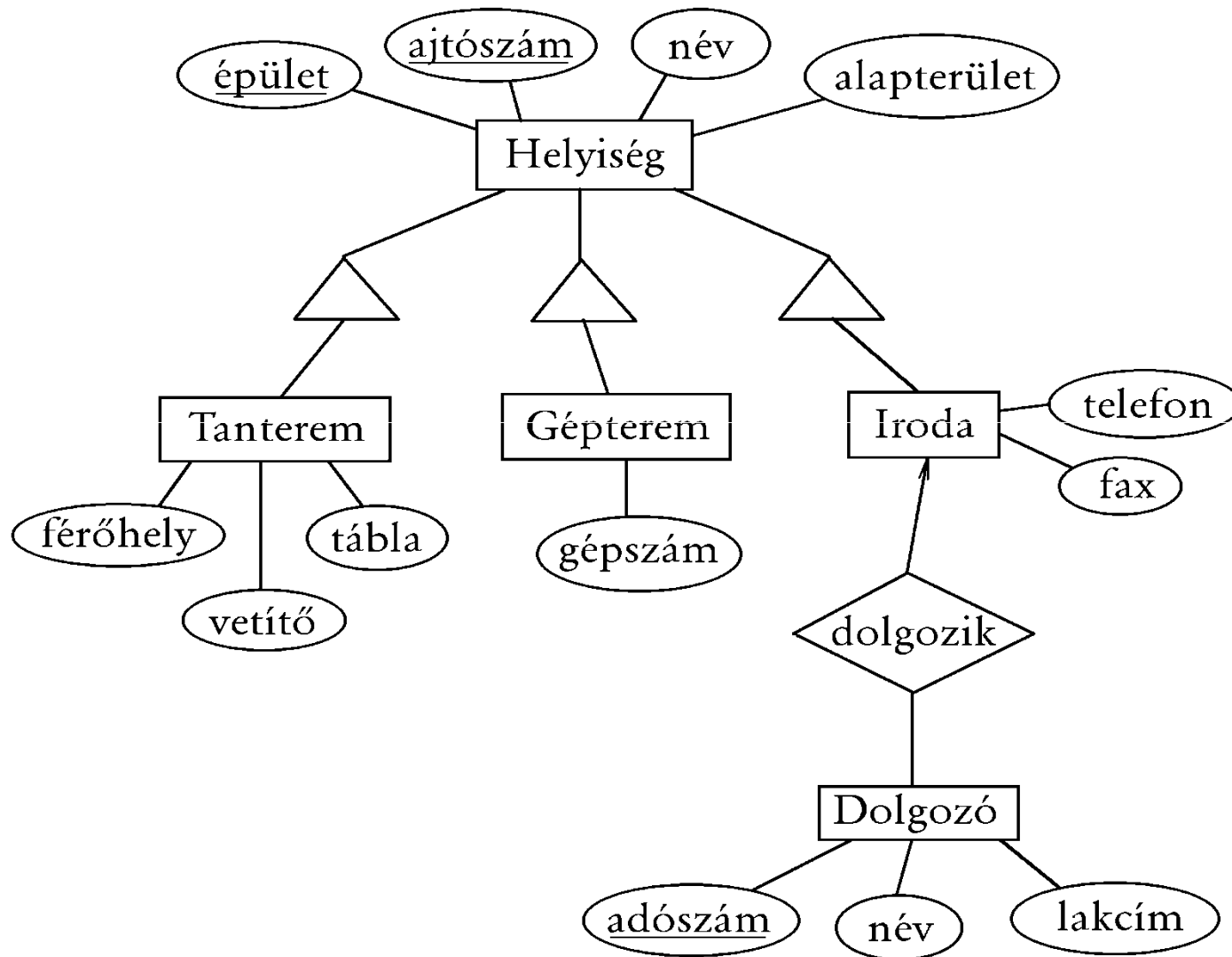
a kulcs, a **KÖLCSÖN** táblát most sem a **KÖNYV**, sem az **OLVASÓ** táblába nem tudjuk beolvasztani. Az adatbázisséma ezért a következő:

**KÖNYV** (könyvszám, szerző, cím)

**OLVASÓ** (olvasószám, név, lakcím)

**KÖLCSÖN** (könyvszám, olvasószám, kivétel, visszahozás)

# Specializáló kapcsolatok leképezése



# Specializáló kapcsolatok leképezése

1. **Minden altípushoz külön tábla felvétele**, egy egyed csak egy táblában szerepel. Az altípusok öröklik 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. sz. terem alapterületét keressük).
- Kombinált altípus (például számítógépes tanterem) csak új altípus felvételével kezelhető.

# Specializáló kapcsolatok leképezése

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 öröklik.

(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:** Itt is előfordulhat, hogy több táblában kell keresni (például ha a tantermek nevére és férőhelyére vagyunk kíváncsiak).

# Specializáló kapcsolatok leképezése

3. **Egy közös tábla felvétele, az attribútumok úniójával.** 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.
- Elveszítjük a típusinformációt (például ha a gépteremnél a gépszám nem ismert és ezért NULL, akkor a gépterem lényegében az egyéb helyiségek kategóriájába kerül).