

# 4.előadás / 1.része: Adatbázisok-I.

dr. Hajas Csilla (ELTE IK) (2020)

<http://sila.hajas.elte.hu/>

Az előadások az SQL gyakorlatokat támogatják, ezért a Tankönyvet nem lineárisan dolgozzuk fel, más a sorrend

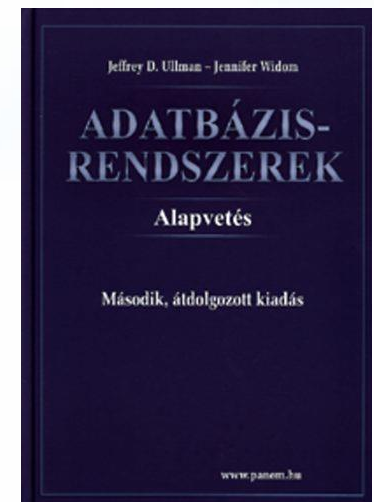
**4.előadás az 1.előadás folytatása:**

Adatmodellezésre feladatok, E/K modell, E/K diagram leképezése relációsémákra

A mai témakörök a Tankönyvben:

4.1-4.4. E/K-modell/diagram elemei

4.5-4.6. E/K-diagram átírása relációkra

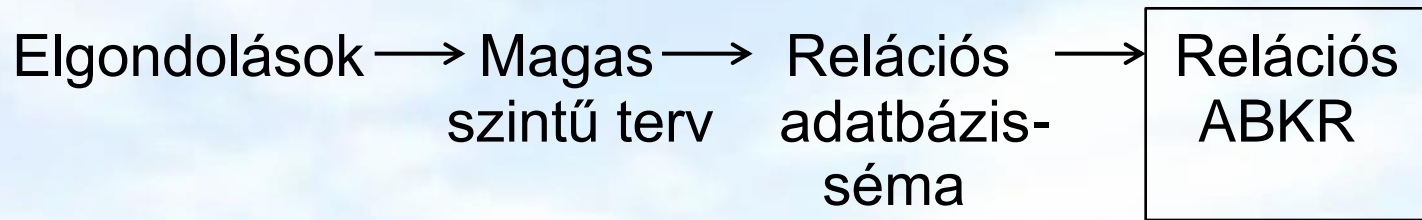


# [ismétlés] Mi az adatmodell?

- Az adatmodell a valóság fogalmainak, kapcsolatainak, tevékenységeinek magasabb szintű ábrázolása
- **Kettős feladat:** az adatmodell megadja, hogy a számítógép számára és a felhasználó számára hogy néznek ki adatok.
- **Az adatmodell:** adatok leírására szolgáló jelölés. Ez a leírás általában az alábbi három részből áll:
  1. **Az adat struktúrája** (struktúrák, tömbök, rekordok, fizikai és fogalmi adatstruktúrák, hamarosan példákat is mutatunk)
  2. **Az adaton végezhető műveletek** (lekérdezések, módosítások, feldolgozások legyenek megfogalmazhatók és hatékonyan legyenek implementálva)
  3. **Az adatokra tett megszorítások** (milyen adatokat engedélyezünk, milyen megszorításokat teszünk?)

# [ismétlés] Adatbázis-modellezés

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

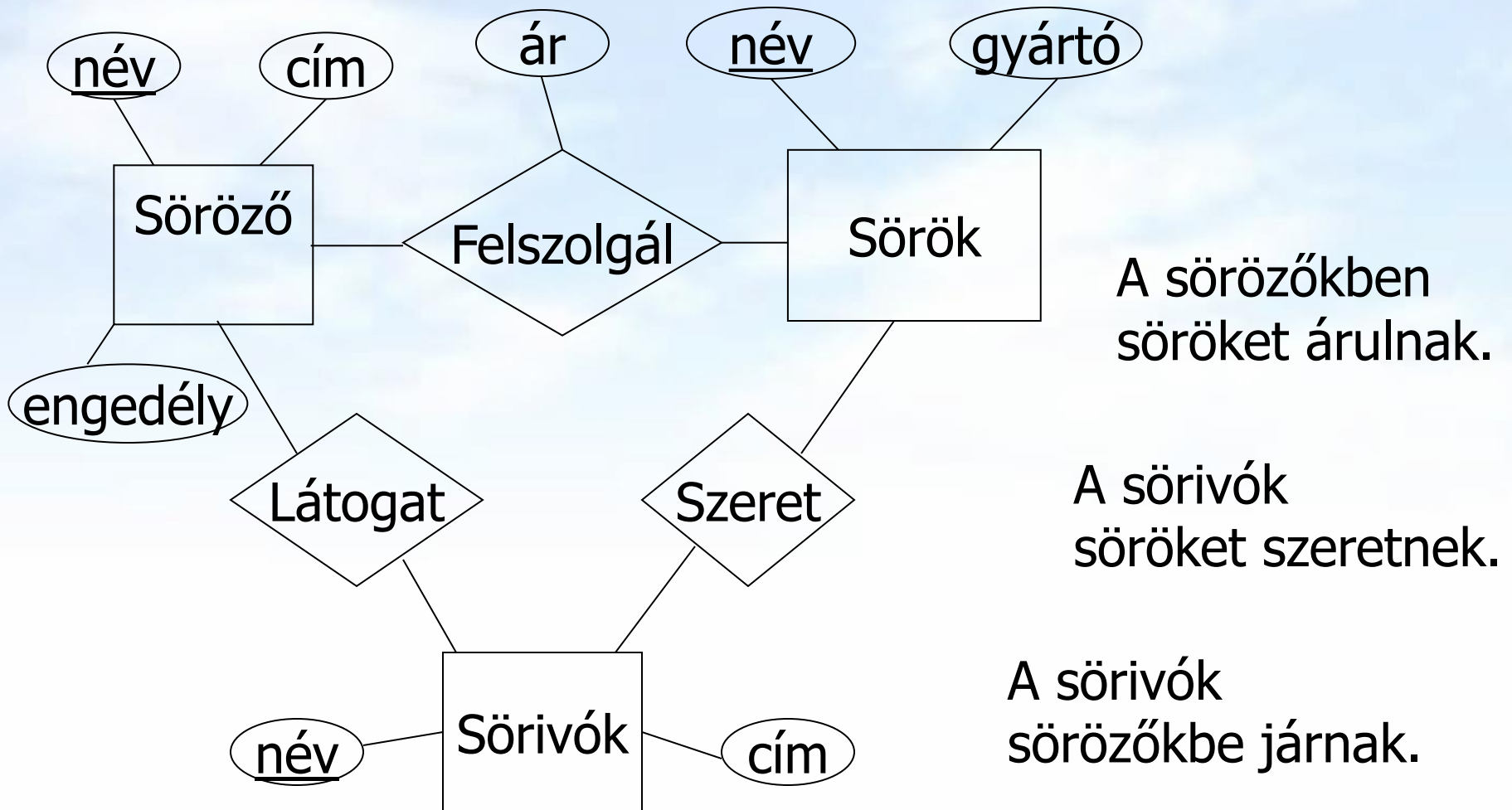


- Modellezés
  - komplex valós világ leképezése, absztrakció
- **Tervezési fázis:**
  - Milyen információkat kell tárolni?
  - Mely információelemek kapcsolódnak egymáshoz?
  - Milyen megszorításokat kell figyelembe venni?

# [ismétlés] Egyed-kapcsolat modell

- **Egyed-kapcsolat modell (E/K diagram elemei)**  
(Entity-relationship ER) alapfogalmak:
- **Egyedhalmazok (jelölés: téglalap)**
  - Miről gyűjtünk adatokat?
  - Mit tegyünk egy gyűjteménybe? - hasonlóság
  - Hasonló egyedek összessége
- **Attribútumok (jelölés: ovális)**
  - Megfigyelhető tulajdonságok, megfigyelt értékek
  - Az egyedek tulajdonságait írják le
- **Kapcsolatok (jelölés: rombusz +vonalak v. nyilak)**
  - Más egyedhalmazokkal való kapcsolatuk

# [emlékeztető] Az előadás példája (E/K)



# Példa: E/K diagram átírása relációkká

## Az egyedek és sok-sok kapcsolatok átírása

- Az egyedhalmazok átírása  
(aláhúzás jelöli a kulcs attribútumokat)

**Sörök(név, gyártó)**

**Sörözők(név, város, tulaj, engedély)**

**Sörivők(név, város, tel)**

- Sok-sok kapcsolatok átírása

**Szeret(név, sör)**

**Felhasználó(söröző, sör, ár)**

**Látogat(név, söröző)**

# Példa: E/K diagram átírása relációkká sok-egy kapcsolatok átírásának 1.lépése



A kapcsolatoknak megfelelő sémákban az oszlopokat **átneveztük**, például **a szerepek** alapján, (név,név) séma nem szerepelhetne.

# Példa: E/K diagram átírása relációkká

## sok-egy kapcsolatok átírásának 2.lépése

### Az összevonás (ésszerűsítés)

- 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 Akedvenc(név, sör) összevonható, és kapjuk az új Sörivók(név, cím, AkedvencSöre) sémát.
- Hasonlóan a Dolgozók táblába összevonható az Osztállyal illetve önmagával való kapcsolat

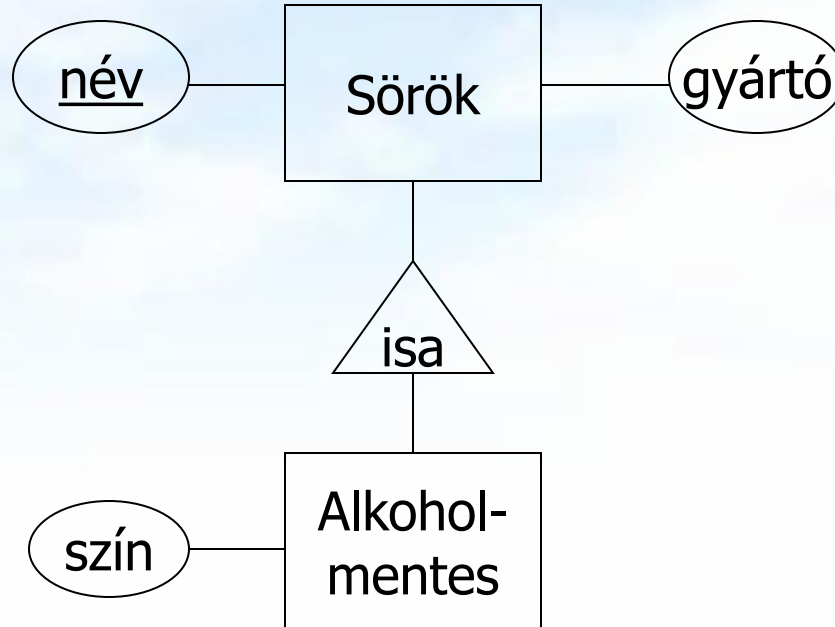


# Ezek voltak az 1.előadáson, folytatás

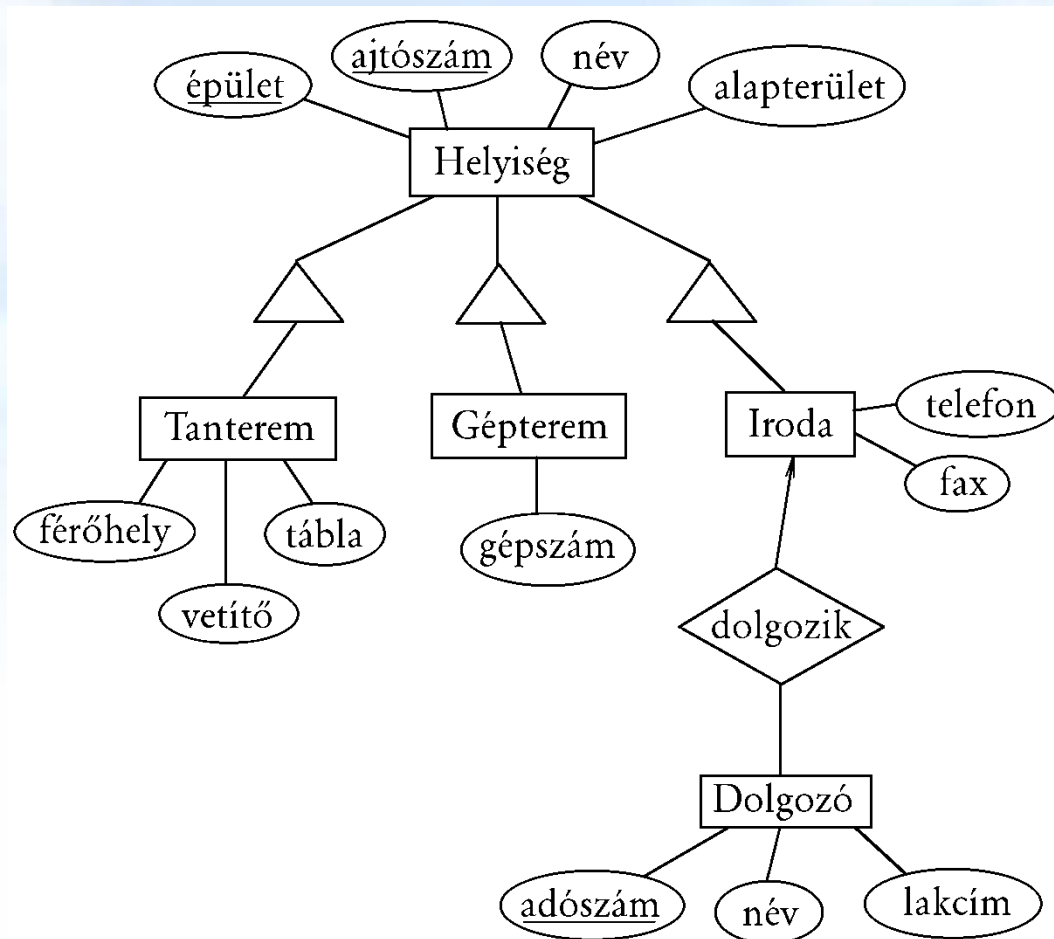
- Alosztályok és öröklődés
  - Speciális „is-a” (az-egy) kapcsolat
- Kulcsok és további megszorítások
- Gyenge egyedhalmazok kezelése
  
- E/K diagram átalakítása relációsémákra
- Osztályhierarchia átalakítása relációkká

# Alosztályok és öröklődés

## Speciális „is-a” (az-egy) kapcsolat

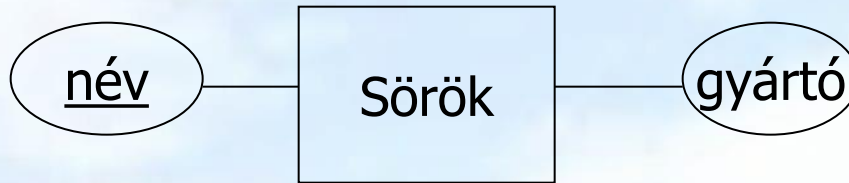


# Példa: „is-a” (az-egy) kapcsolatra

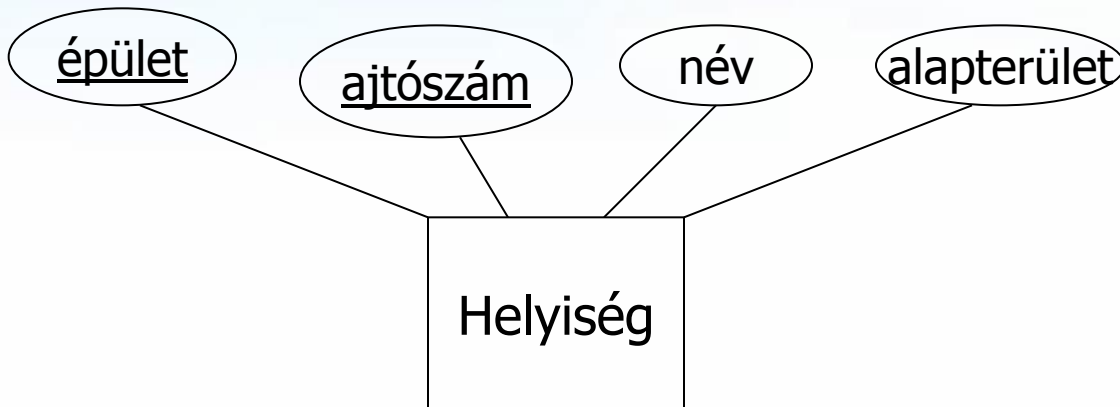


# Kulcs megszorítás jele: aláhúzás

Példa egyszerű kulcsra: név a Sörök elsődleges kulcsa:



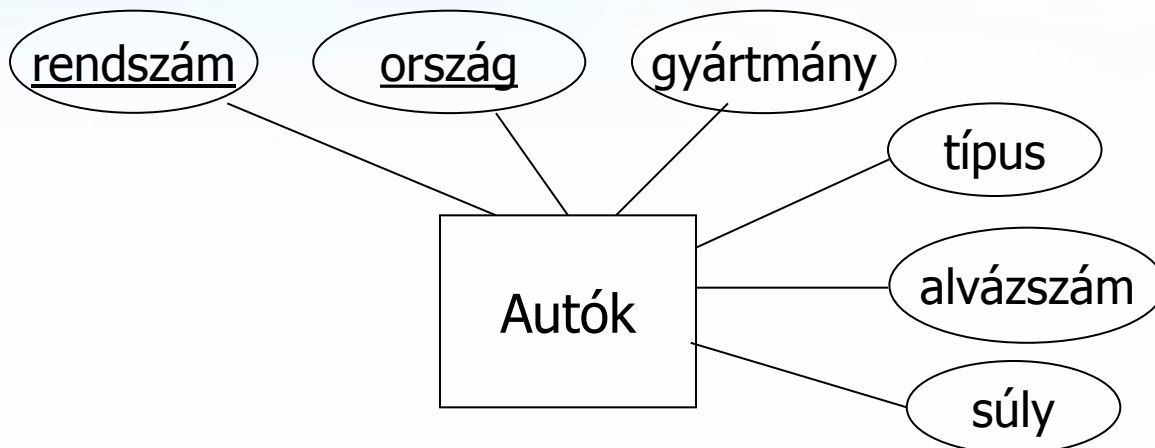
Példa összetett kulcsra: épület, ajtószám két-attribútumos elsődleges kulcsa a Helyiség-nek:



# Kulcs (minimális szuperkulcs)

Példa kulcsra: rendszer, ország

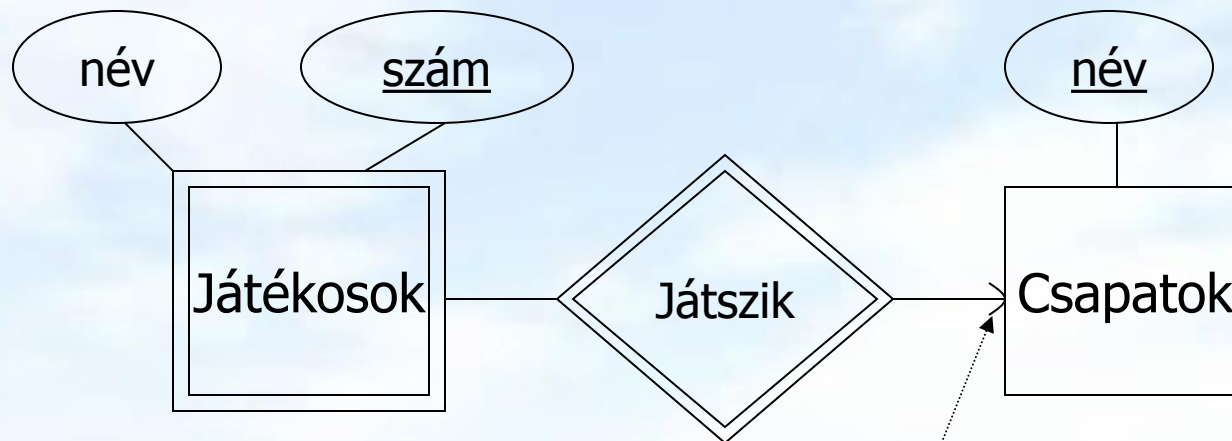
Autók egyedhalmazt leíró tulajdonságok közül több kulcs is van, de egy elsődleges kulcsot kiválasztunk és csak azt jelöljük aláhúzással. Például egyszerű kulcs az alvászám (ez is egyértelműen azonosítja), vagy összetett kulcs rendszer, ország (minimális, vagyis nincs valódi részhalmaza, ami azonosítaná).



# Hivatkozási épség megszorítás jele a kerek végződés —)



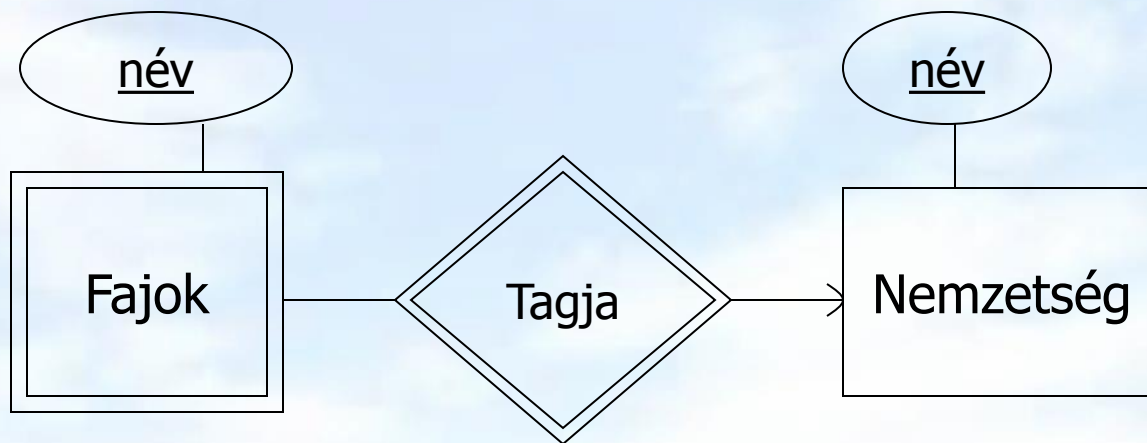
# Erős és gyenge egyedhalmaz



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] A fajokat nemzetségükkel és fajukkal jelölik, például az emberek a **Homo sapiens** fajhoz tartoznak, ahol **Homo** a nemzetség neve, és a **sapiens** a faj neve, mivel a faj neve nem egyértelmű (két vagy több nemzetségben is lehet ugyanolyan név), egy fajt egyértelműen a nemzetség és a faj neve jelöli.



# 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 egyedhalmaz:**
  - például a `vizsgajegy` osztály helyett `jegy` tulajdonságot használjunk.

# E/K-diagram átírása relációkra

Előtte: Relációs adatmodell (Tankönyv 2.1-2.2.)

E/K-diagram átírása relációsémákra (Tk 4.5.-4.6.)

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

# Gyenge egyedhalmaz átírása



Hostgépek(hostNév, cím)

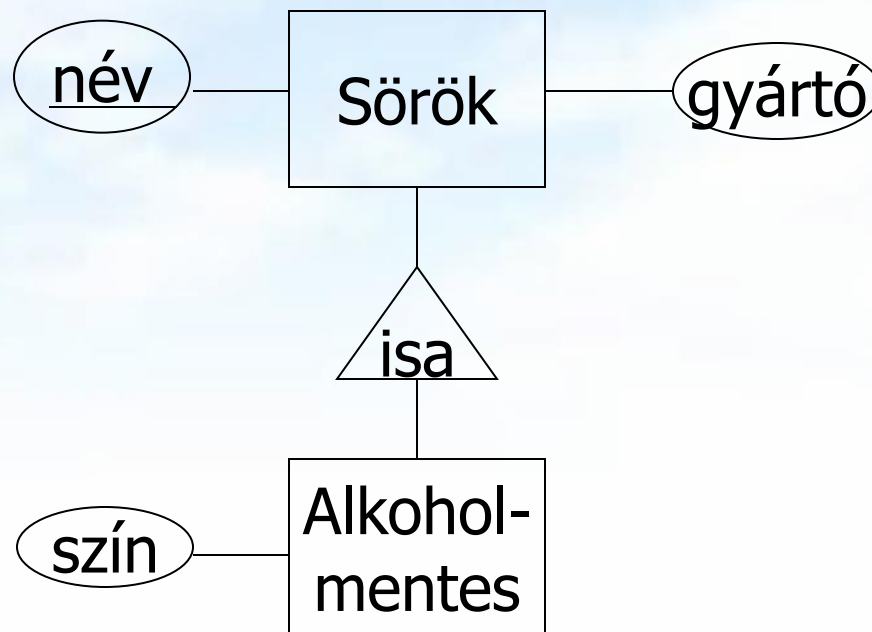
Logins(loginNév, hostNév, jelszó)

~~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

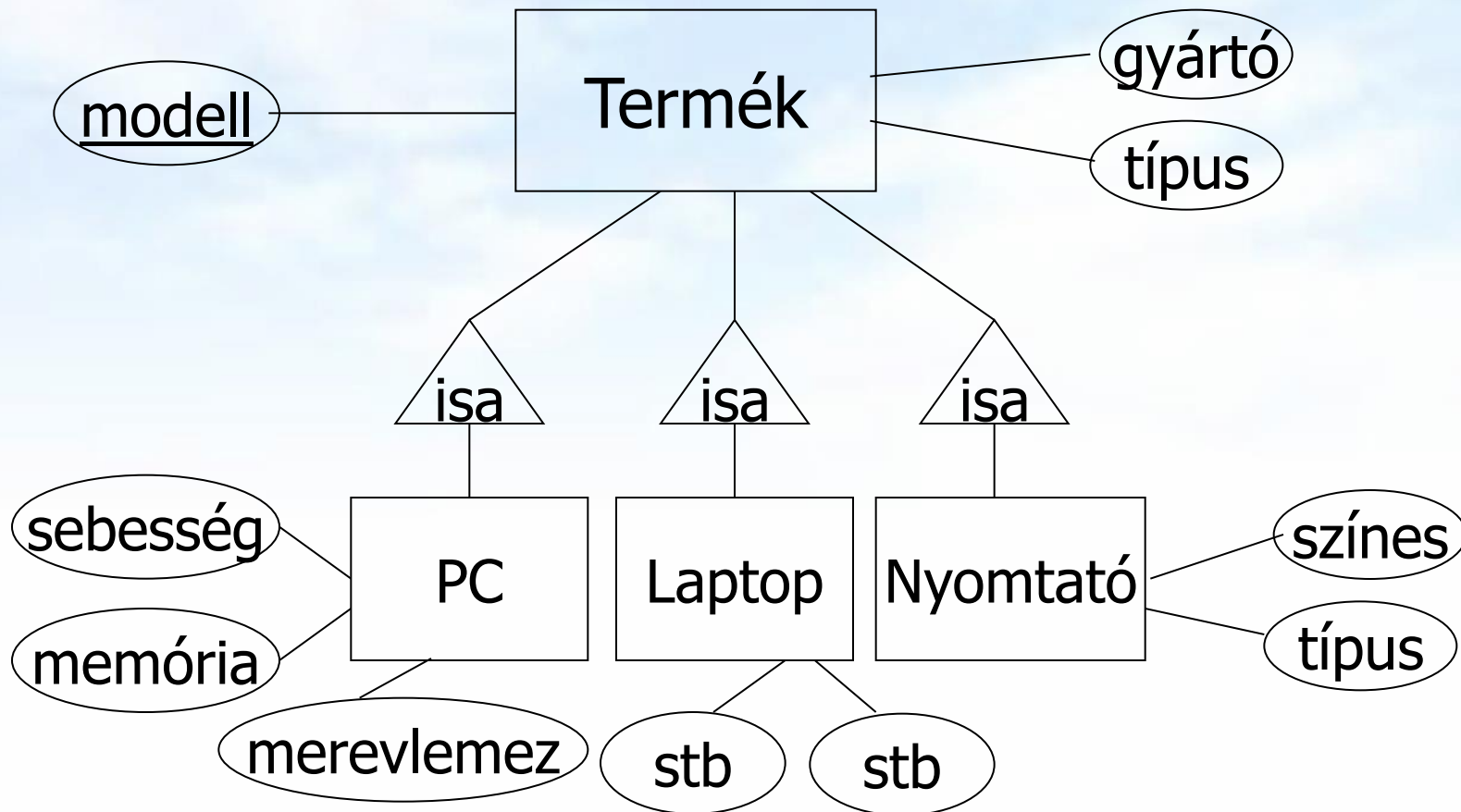
Mindenféle lekérdezésre jó, mivel csak egy tábla van, egy relációsémába uniózzuk az összes attribútumot, ha egy egyed nem rendelkezik egy alosztály speciális tulajdonságával, akkor ide NULL értéket írunk.

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

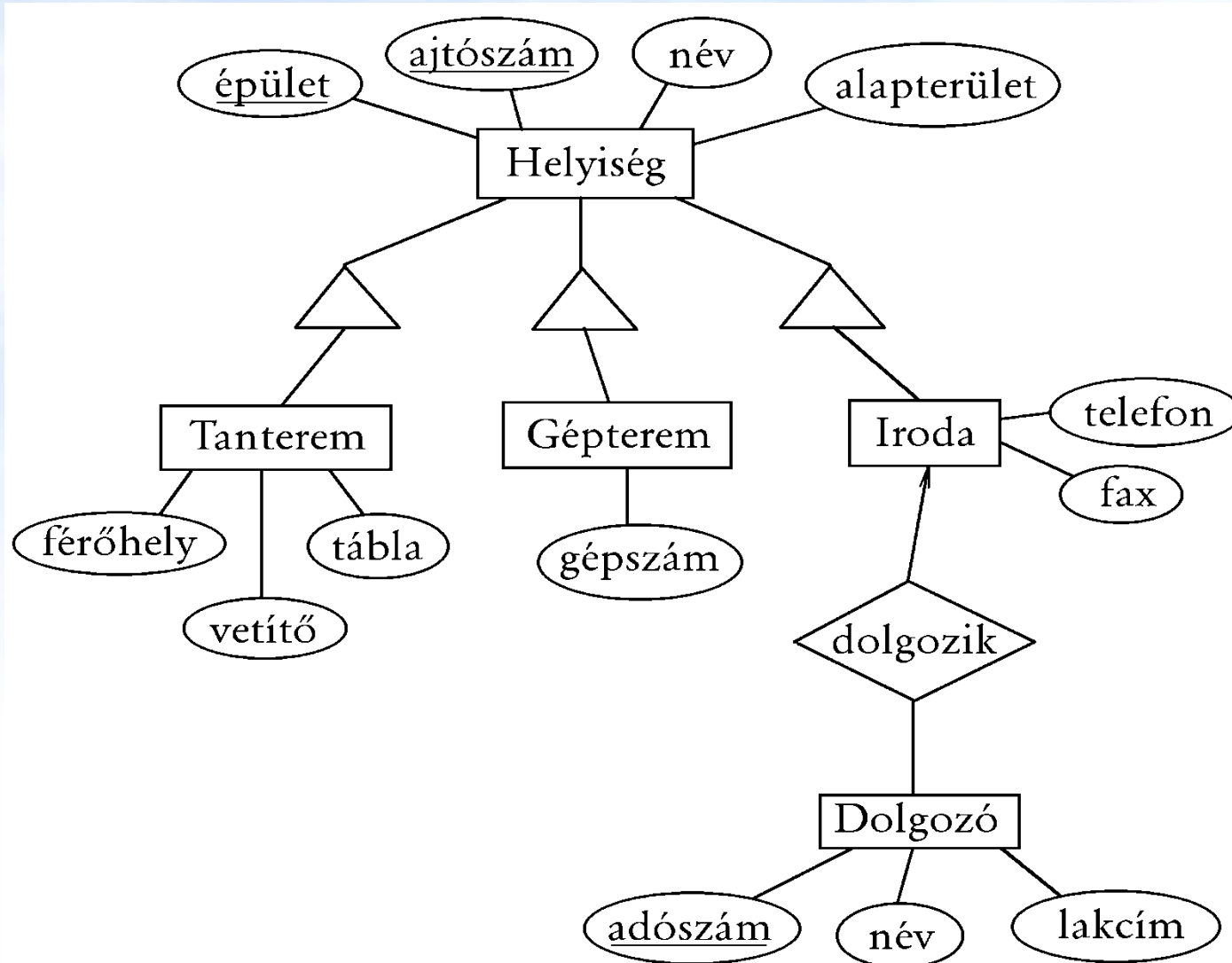


# Példa: Termékek (Tankönyv feladata)

E/K-t írjuk át a három megközelítéssel relációsémákra!



# Feladat: Alosztály átírása relációkká



# Folytatás: SQL2.pdf

- Ezt az 4.előadást is folytatjuk a 2.résszel:  
Hogyan hozzuk létre a táblákat (rel.séma) és megszorításokat az SQL-ben, lásd [SQL2.pdf](#)