

1.előadás: Adatbázisok

dr. Hajas Csilla (ELTE IK)

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

Az előadások az SQL gyakorlatokat támogatják, ezért a Tankönyvet nem lineárisan dolgozzuk fel (1.fej.később)

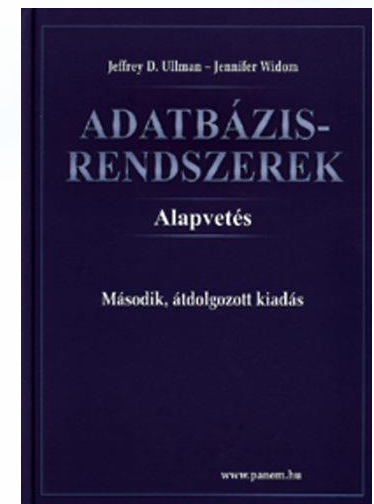
Bevezetés/Tervezés: táblák előkészítése
Adatmodellezésre feladatok, E/K modell,
E/K diagram leképezése relációsémákra

A mai témakörök a Tankönyvben:

2.1.-2.2. Adatmodellek, relációs modell

4.1.-4.4. E/K-modell elemei

4.5.-4.6. E/K-diagram átírása relációkra



Bevezető példa (reláció = tábla)

- Az adat, adatbázis, adatbázis-kezelő rendszer (software) alapfogalmakat később vesszük (Tk.1.fej.--> 10.előadás)
- Naponta találkozunk adatbázisokkal
 - 1960-as évektől a korai DBMS: banki rendszerek, repülőgép-helyfoglalás, vállalati nyilvántartások
 - Napi szinten: Google, Yahoo, Amazon.com, egyetemi tanulmányi rendszerek, Neptun
- **1.példa: a jelenléti ív** (az első előadás papíros példája)
- Reláció = tábla = séma/szerkezet + előfordulás/tartalom
- Séma = a reláció szerkezetének leírása (tábla fejléce)
- Előfordulás v. példány = a tábla sorai, adott időpontban a tábla aktuális tartalma dinamikusan változik (példában szereplő tábla tartalma a tantárgyfelvétel időszakában)

Bevezetés: Ki ismeri az SQL-t?

- **Ki ismeri az SQL-t?** Van-e különbség?

Tetszőleges táblát lekérdezve megegyezik-e az eredmény?

```
(1) SELECT B FROM R
    WHERE A < 10 OR A >= 10;
```

```
(2) SELECT B FROM R;
```

- **Itt mi a helyzet ezzel?**

```
(3) SELECT A FROM R, S
    WHERE R.B = S.B;
```

```
(4) SELECT A FROM R
    WHERE B IN
    (SELECT B FROM S);
```

Egy példa relációra:

R	A	B
	5	20
	10	30
	20	40

Reláció a.) séma: R(A,B)

b.) előfordulás (sorok hz)

A lekérdezésekben csak a rel.sémák szerepelnek, itt pl.: R(A,B), és S(B,C)

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?)

A fontosabb adatmodellek

- **Hálós, hierarchikus adatmodell** (gráf-orientált, fizikai szintű, ill. apa-fiú kapcsolatok gráfja, hatékony keresés)
- **Relációs adatmodell** (táblák rendszere, könnyen megfogalmazható műveletek), magában foglalja az **objektumrelációs kiterjesztést** is (strukturált típusok, metódusok), SQL/Object, SQL/CLI, SQL/PSM (PL/SQL)
- **Objektum-orientált adatmodell** (az adatbázis-kezelés funkcionalitásainak biztosítása érdekében gyakran relációs adatmodellre épül), ODMG: ODL és OQL
- **Logikai adatmodell** (szakértői rendszerek, tények és következtetési szabályok rendszere)
- **Dokumentum típusú adatok, félig-strukturált adatmodell** (XML-dokumentum)
- **Gráf adatbázisok, stb.** (további adatmodellek)

Példa relációs adatmodellre

Az adatbázis sémája: **Sör** (név, gyártó),
Söröző (név, város, tulaj),
Felszolgál (sör, söröző, ár).



Példa féligstrukturált adatra (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<söröző típus="étterem">
```

```
  <név>Makk 7-es</név>
```

```
  <város>Budapest</város>
```

```
  <tulaj>Géza</tulaj>
```

```
  <telefon>+36-70-123-2345</telefon>
```

```
  <telefon>+36-70-123-2346</telefon>
```

```
</söröző>
```

```
<söröző típus="kocsma">
```

```
  <név>Lórúgás</név>
```

```
  <város>Eger</város>
```

```
  <telefon>+36-30-451-1894</telefon>
```

```
</söröző>
```

Relációs adatmodell története

- **E.F. Codd** 1970-ben publikált egy cikket
A Relational Model of Data for Large Shared Data Banks
Link: <http://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf>
amelyben azt javasolta, hogy az adatokat táblázatokban, **relációkban** tárolják. Az elméletére alapozva jött létre a relációs adatmodell, és erre épülve jöttek létre a relációs adatmodellen alapuló relációs adatbázis-kezelők.
- **Relációs** (objektum-relációs) **adatbázis-kezelők** például:
ORACLE , INFORMIX , SYSDATABASE , INGRES, DB2, stb
- Adatbázisok-1 gyakorlaton **ORACLE** adatbázis-kezelő rendszert használunk.

Relációs séma

- Adatok gyűjteményét kezeli (gyűjtemény azonosítása: név)
A gyűjtemény - **R reláció** (tábla, táblázat) megadása
- A gyűjtemény milyen típusú adatokat gyűjt?
adattípus: **sor-típus**. A sor-típus (egy n-es) megadása:
<Attribútumnév₁: értéktípus₁, ... , Attrnév_n: értéktípus_n>
röviden **<A₁, ... , A_n>**
- **Relációséma**: Relációnév (sortípus) (itt: kerek zárójelben!)
vagyis **R(Anév₁: értéktípus₁, ... , Anév_n: értéktípus_n)**
röviden **R(A₁, ... , A_n)** ill. **U = {A₁, ... , A_n}** jelöléssel **R(U)**
- A gyakorlatban SQL DDL create table utasításával megadjuk a relációsémát, oszlopneveket és típusukat, továbbá később lesznek még megszorításokat (pl. kulcs).

Reláció előfordulás

- **Relációs séma feletti reláció előfordulás (példány, instance)**
A sor-típusnak megfelelő véges sok sor (sorok halmaza)
 $\{t_1, \dots, t_m\}$ ahol t_i (tuple, sor, rekord) $i = 1, \dots, m$ (véges sok)
- Mit jelent egy konkrét sor? **sor $\langle A_1: \text{érték}_1, \dots, A_n: \text{érték}_n \rangle$**
 $t_i = \langle v_{i1}, \dots, v_{in} \rangle$ (vagyis egy sor n db értékből áll)
 m - számosság (sorok száma)
 n - dimenzió (attribútumok száma)
- **Értéktartományok:** A reláció minden A_j attribútumához tartozik egy $\text{dom}(A_j)$ értéktartomány (adott értéktípusú értékek halmaza) (1-normálforma feltétel: atomi típusú)
- v_{ij} ebből a **$\text{dom}(A_j)$** tartományból veszi fel az értékét
vagy speciális NULL érték (hiányzó érték) is lehet!

Logikai szinten: táblázatos szemléltetés

- A relációk táblákban jelennek meg. A tábláknak egyedi neve van. A relációk oszlopait az attribútumok címzik. A tábla sorait tetszőlegesen megcserélhetjük, sorok sorrendje lényegtelen (a halmazszemlélet miatt)

Mivel attribútumok halmazáról van szó, a Példa 1 és Példa 2 relációk nevüktől eltekintve azonosak.

Példa 1

A	B	C
a	b	c
d	a	a
c	b	d

Példa 2

B	C	A
b	c	a
a	a	d
b	d	c

Mivel sorok halmazáról van szó, a Példa 1 és Példa 3 relációk nevüktől eltekintve azonosak.

Példa 3

A	B	C
c	b	d
d	a	a
a	b	c

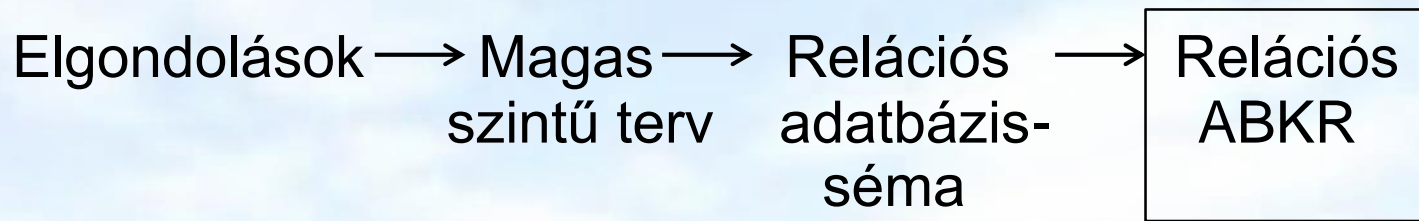
Példa 4

A	B	C
c	b	d
c	b	d
a	b	c

Ebben a modellben Példa 4 nem reláció, de a valóságban megengedünk multihalmazokat lásd később SQL

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?

Egyed-kapcsolat modell elemei

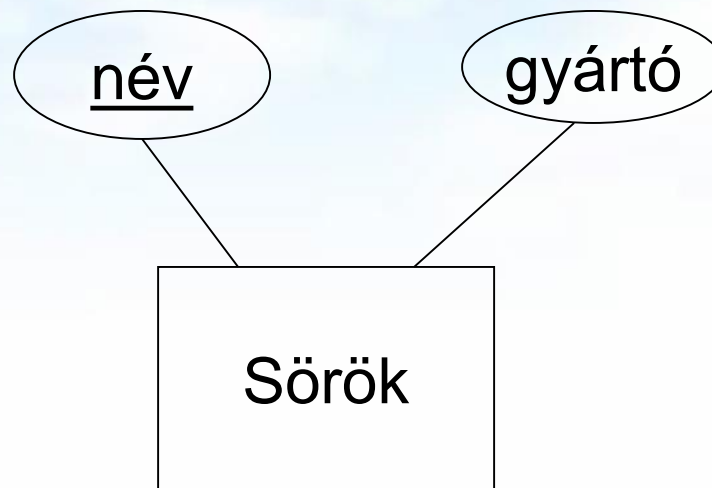
- **Egyed-kapcsolat modell: E/K modell**
(Entity-relationship ER) alapfogalmak:
- **Egyedhalmazok** (absztrakt objektumok osztálya)
 - Miről gyűjtünk adatokat?
 - Mit tegyünk egy gyűjteménybe? - hasonlóság
 - Hasonló egyedek összessége
- **Attribútumok**
 - Megfigyelhető tulajdonságok, megfigyelt értékek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

E/K modell elemei: Egyedhalmazok

- E/K diagram sématervező eszköz, séma-szinten
- $E(A_1, \dots, A_n)$ egyedhalmaz/egyedtípus **séma**:
 - E az egyedhalmaz neve,
 - A_1, \dots, A_n tulajdonságok (attribútumok),
 - $DOM(A_i)$ – lehetséges értékek halmaza
- $E(A_1, \dots, A_n)$ sémájú egyedhalmaz **előfordulása**:
 - A konkrét egyedekből áll
 - $E = \{e_1, \dots, e_m\}$ egyedek (entitások) halmaza, ahol $e_i(k) \in DOM(A_k)$,
 - Az entitások konkrét egyedek léteznek és megkülönböztethetők (vagyis nincs két egyed, amely minden tulajdonságon megegyezne)

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, Jele: **rombusz**
- 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.

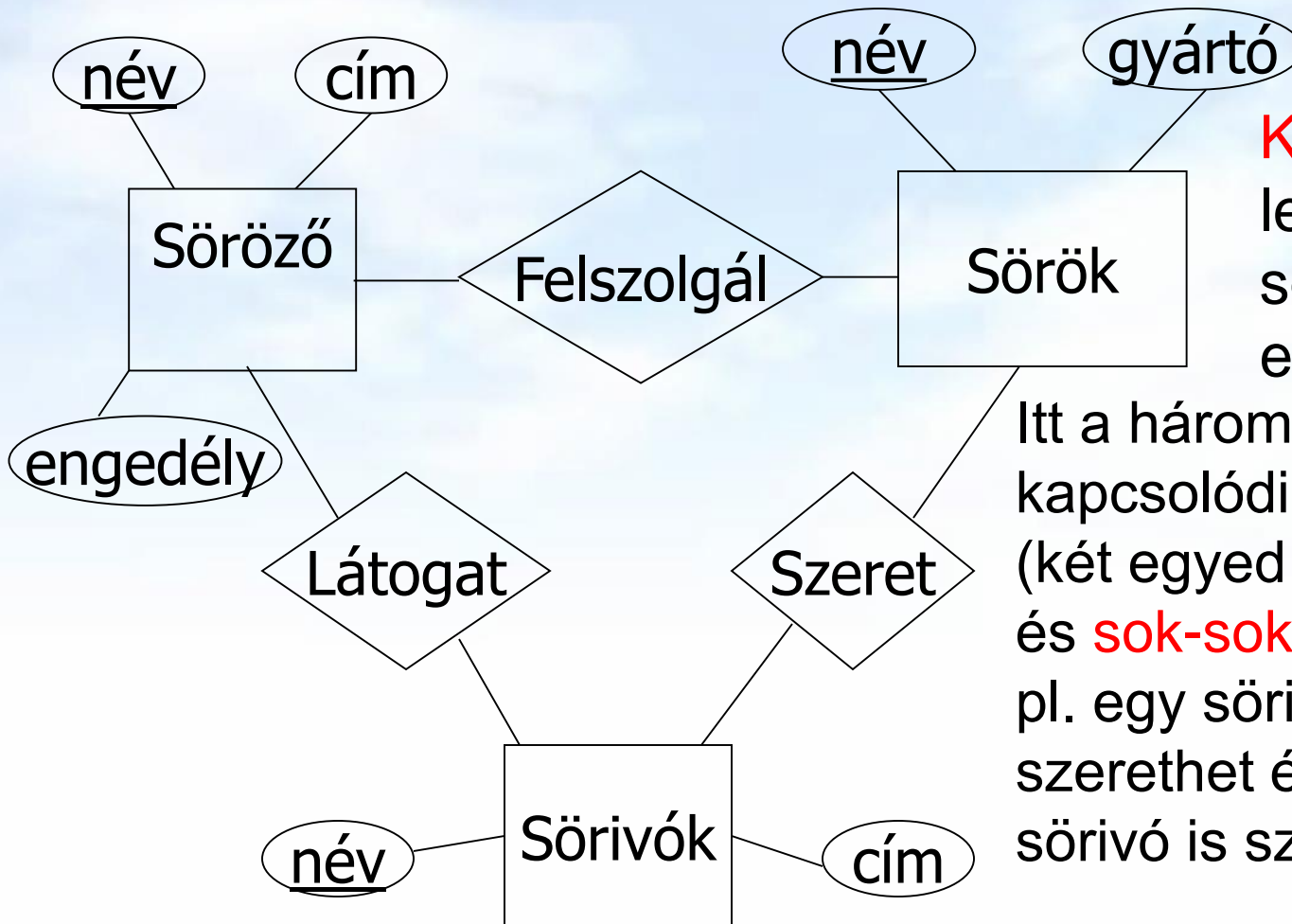
E/K-diagram: Kapcsolatok

- A kapcsolatok jele: **rombusz**



Példa: Bináris kapcsolatok típusai

- A kapcsolatok jele: **rombusz**



Kapcsolat típusa
lehet sok-sok,
sok-egy és
egy-egy kapcsolat

Itt a három egyed páronként kapcsolódik, ez 3 db **bináris** (két egyed közötti) **kapcsolat** és **sok-sok kapcsolat**, mert pl. egy sörivő több sört is szerethet és egy sört több sörivő is szerethet.

Kapcsolatok típusai: sok-egy, sok-sok

(két egyedhalmaz között több kapcsolat is lehet)

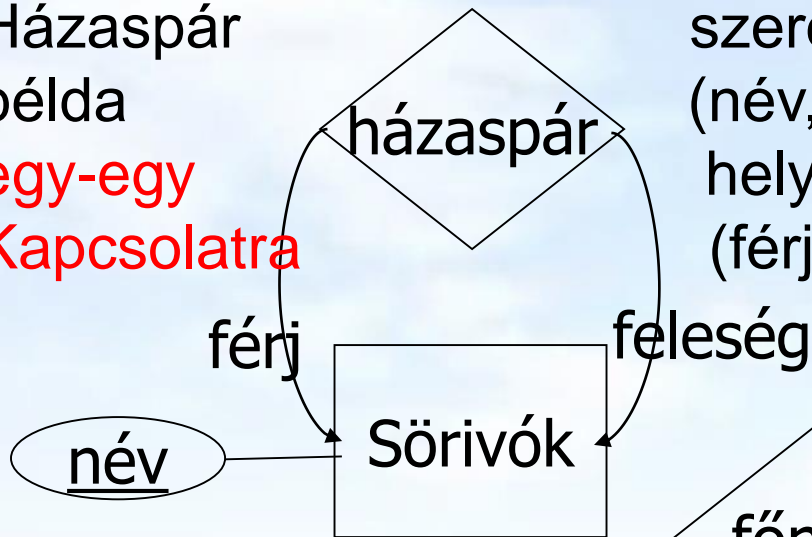


sok-egy kapcsolat:
(nyíl) K előfordulásaiban minden E1-beli egyedhez legfeljebb egy E2-beli egyed tartozhat!

Egy egyedhalmaz önmagával is kapcsolódhat: Szerepek (Roles)

Házaspár
példa

egy-egy
Kapcsolatra



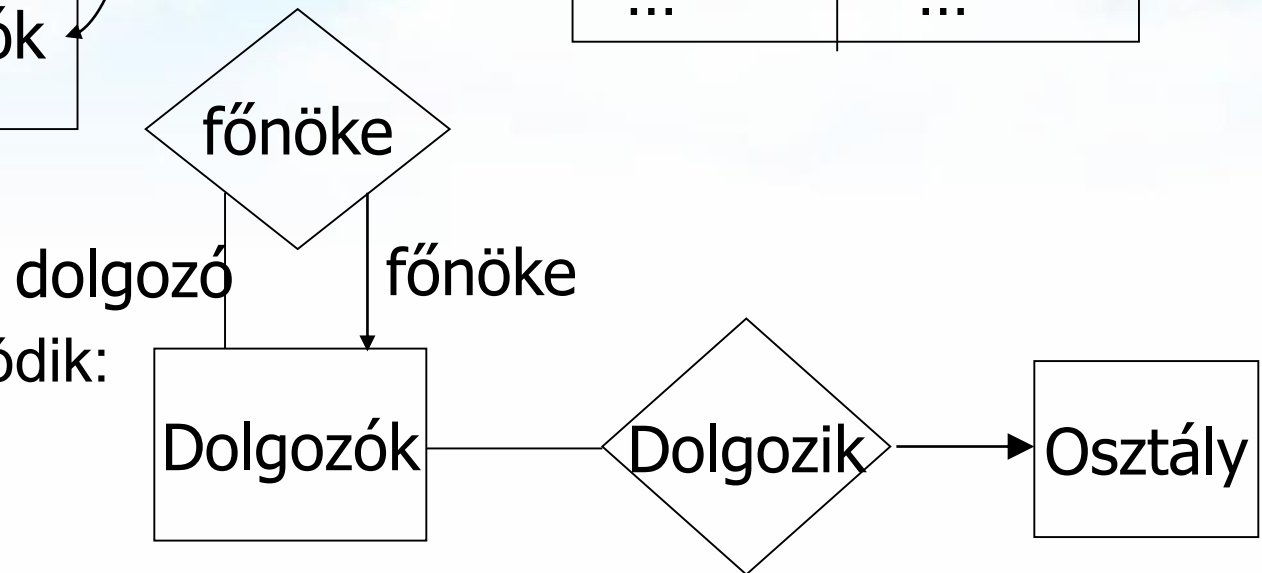
szerepkör
(név,név) pár
helyett lesz
(férj,feleség)

Kapcsolat előfordulás

férj	feleség
Bob	Ann
Joe	Sue
...	...

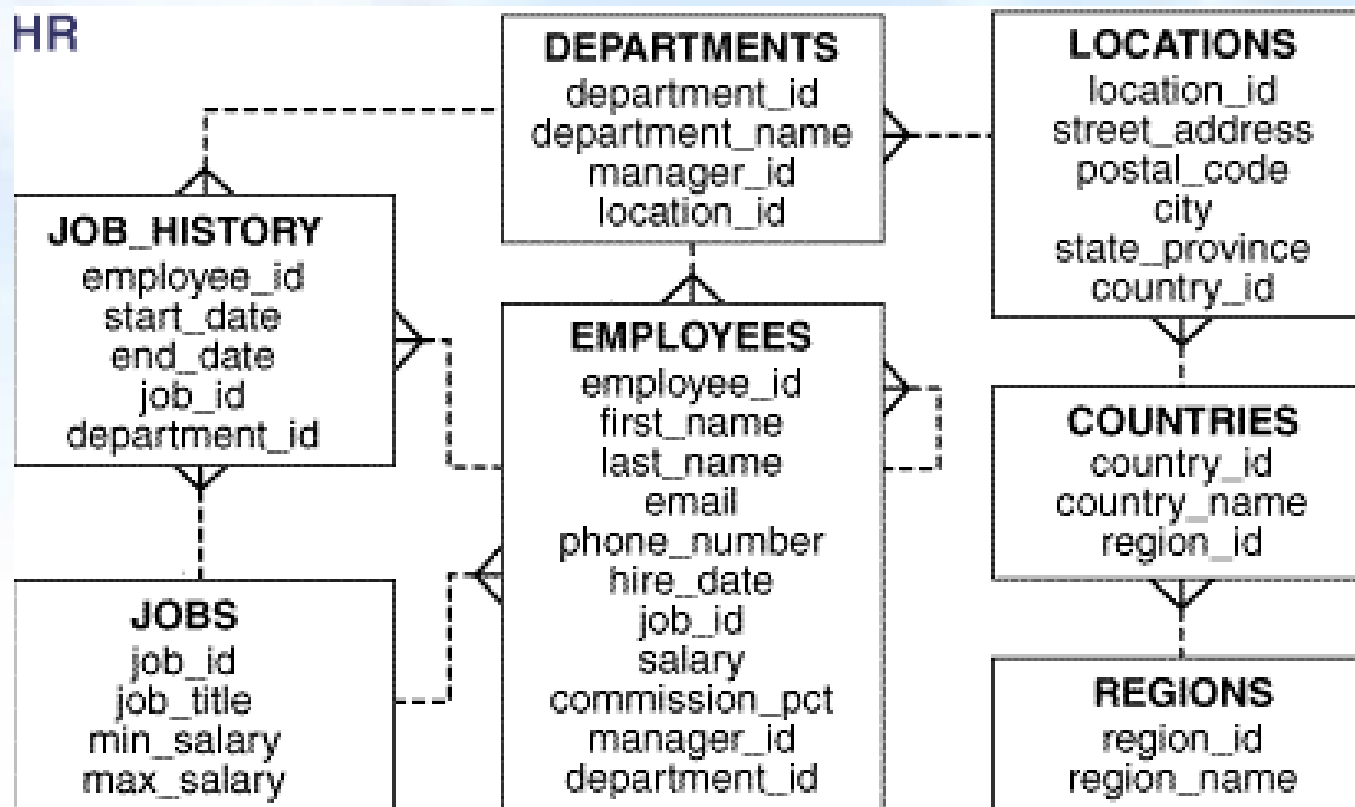
Egy másik példa,
ahol egyedhalmaz
önmagával kapcsolódik:
dolgozó - főnöke

sok-egy kapcsolat

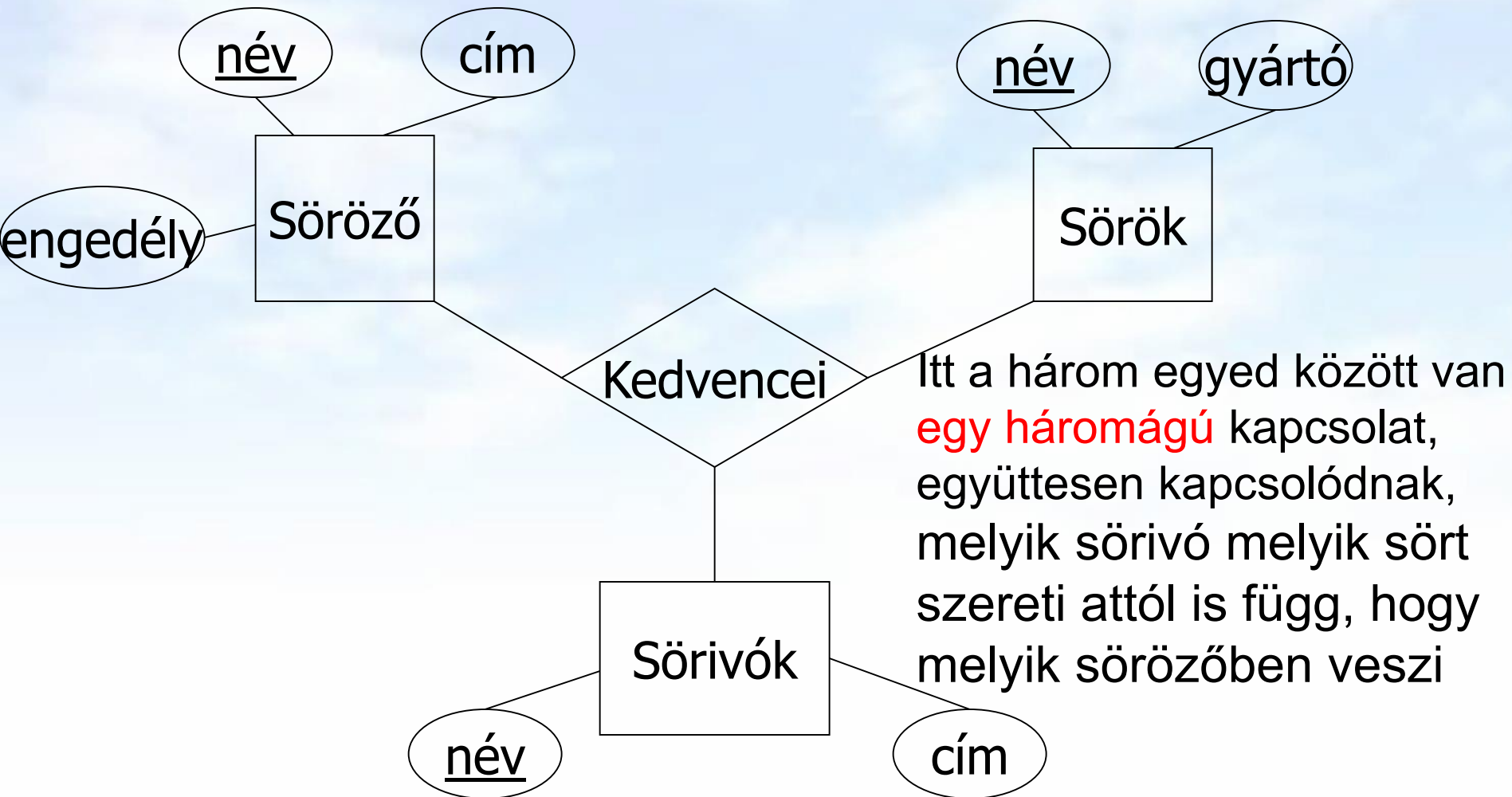


Példa a gyak-on: Oracle HR séma táblái

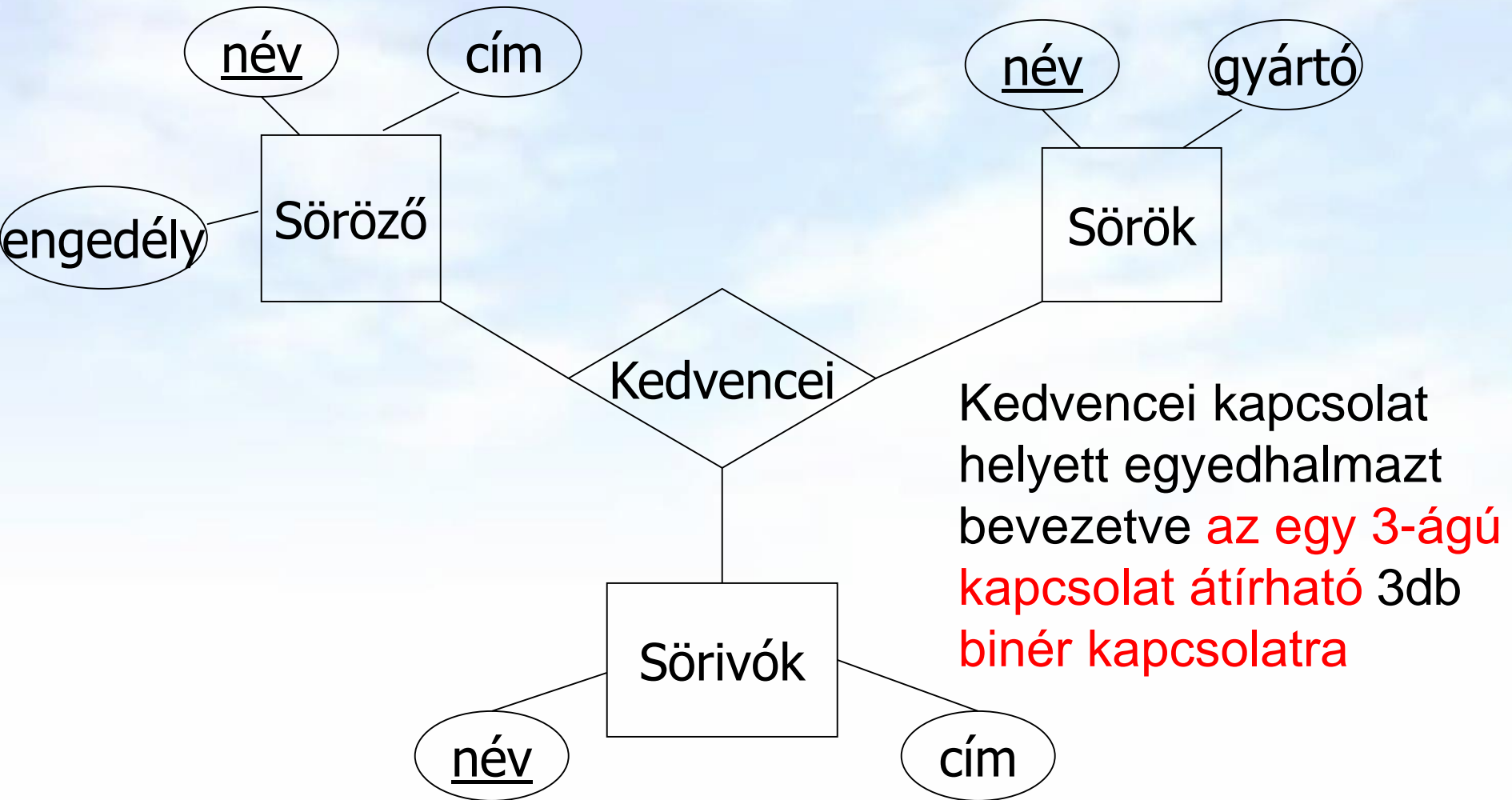
Itt új jelöléssel csak bináris kapcsolatok lehetnek!



Példa: Többágú (3-ágú) kapcsolatra



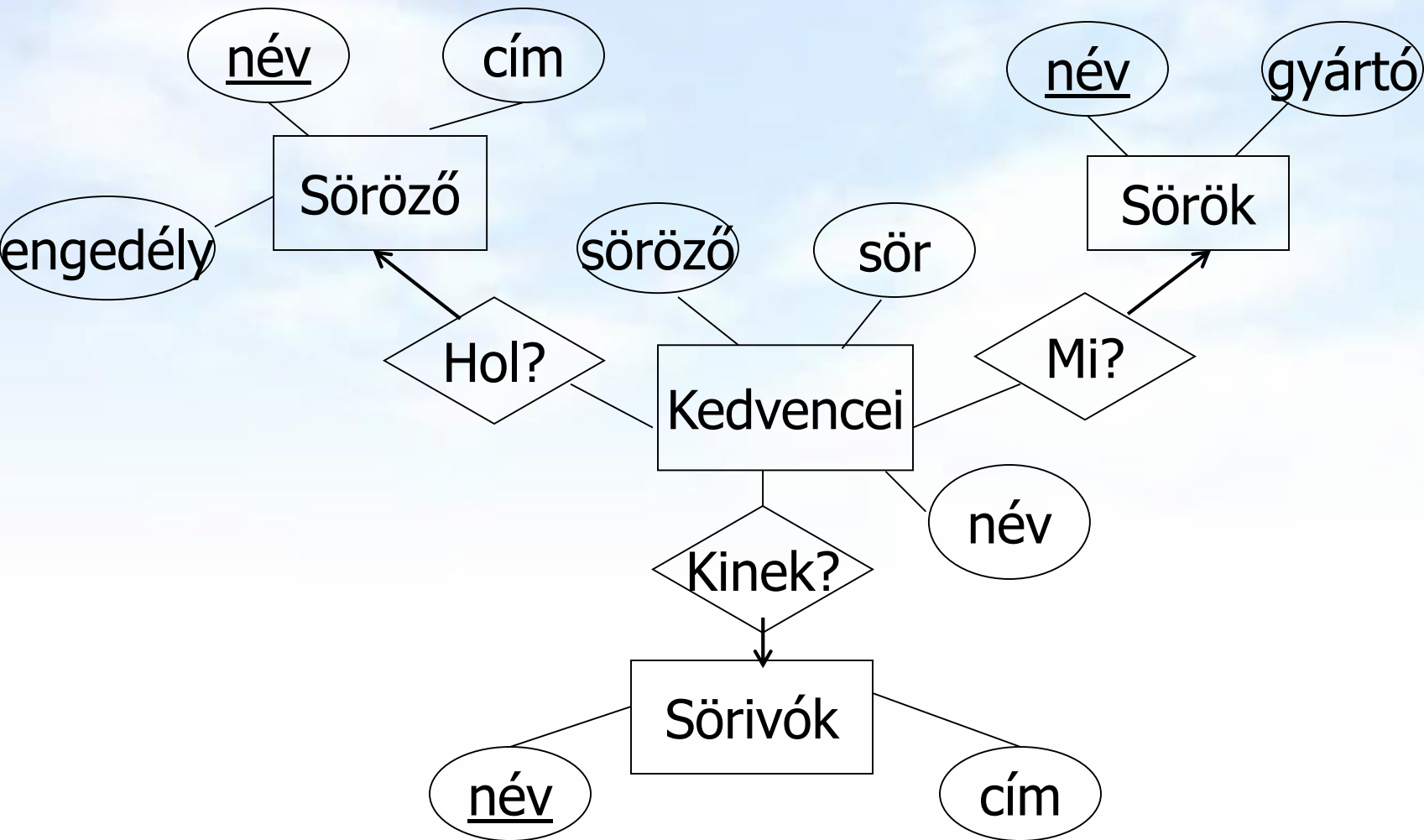
Példa: Átírható-e bináris kapcsolatra?



Példa: Többágú kapcsolat átírása bináris kapcsolatokra

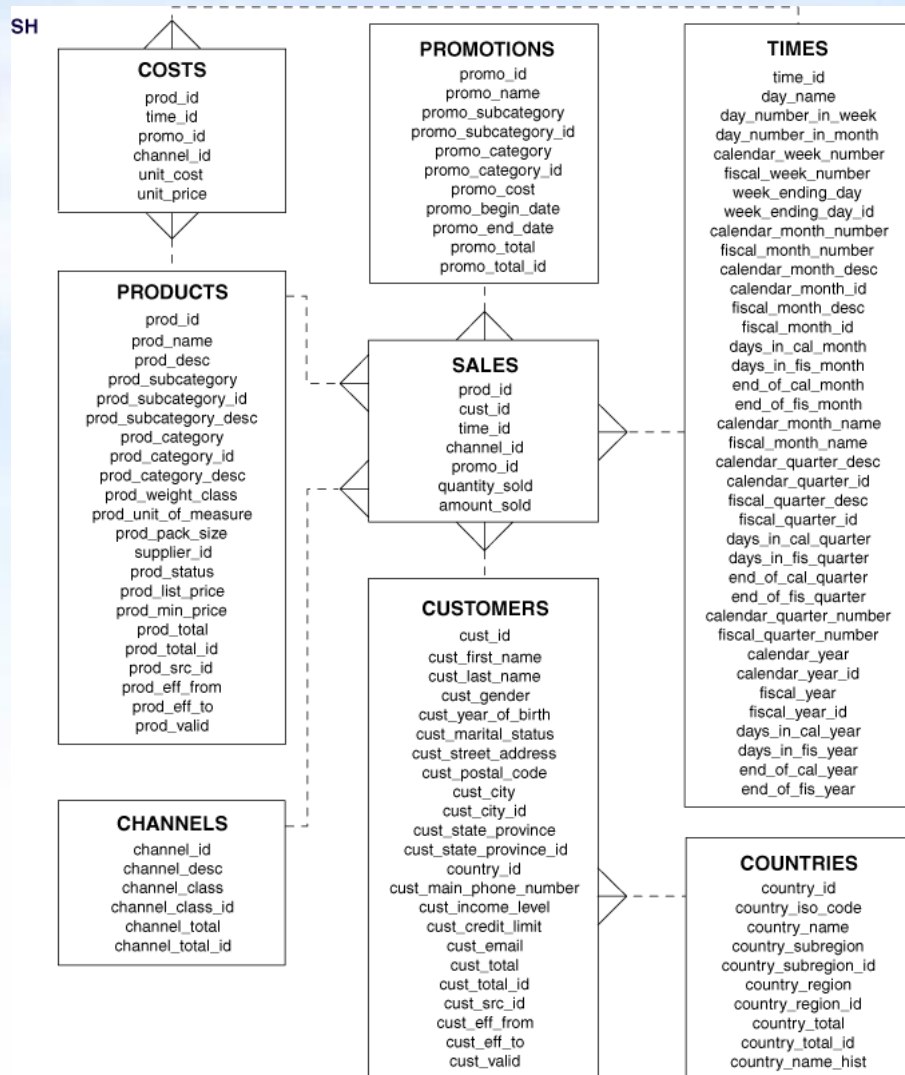
- Kedvencei kapcsolat helyett egyedhalmazt bevezetve ez **a 3-ágú kapcsolat átírható 3db binér kapcsolatra**
- Vagyis Kedvencei kapcsolat helyett egy új egyedhalmazt vezetünk be, aminek három attribútuma van: a név, söröző, sörivó
- és ezt az új egyedhalmazt középen téglalappal jelöljük, és ez kapcsolódik páronként a három egyedhez bináris kapcsolattal, lásd köv lapon:

Példa: Többágú kapcsolat átírása bináris kapcsolatokra



Példa a gyak-on: Oracle SH séma táblái

Itt új jelöléssel csak bináris kapcsolatok lehetnek!



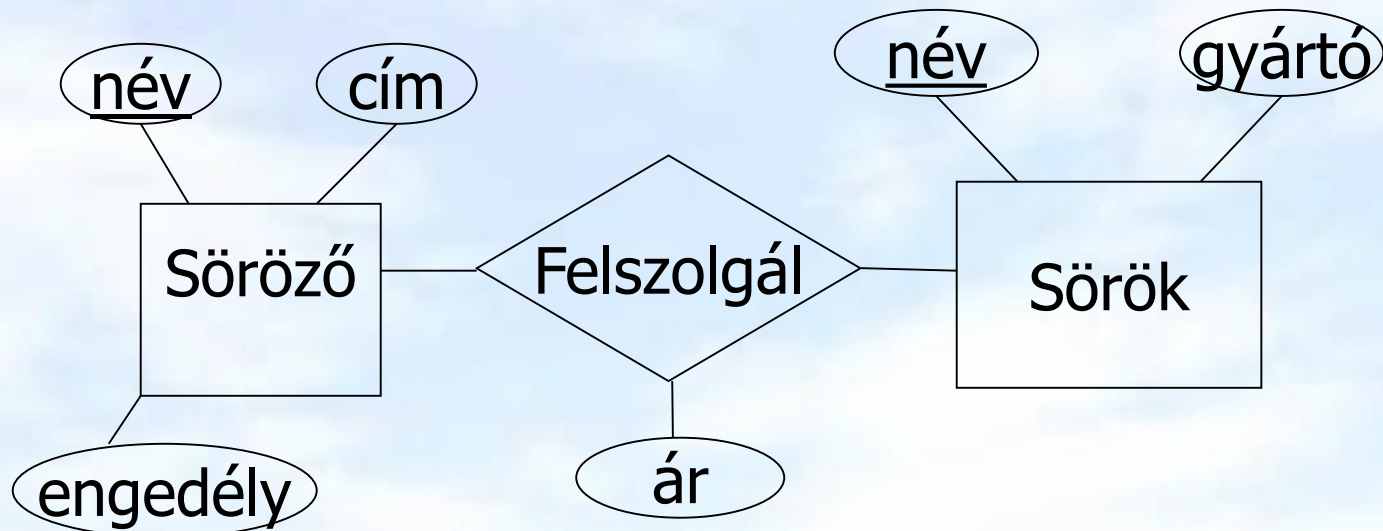
Többágú kapcsolat lett itt is feloldva, mint előbb!

E/K jelölésben középen 5-ágú kapcsolat lenne a SALES vagyis ELADÁS



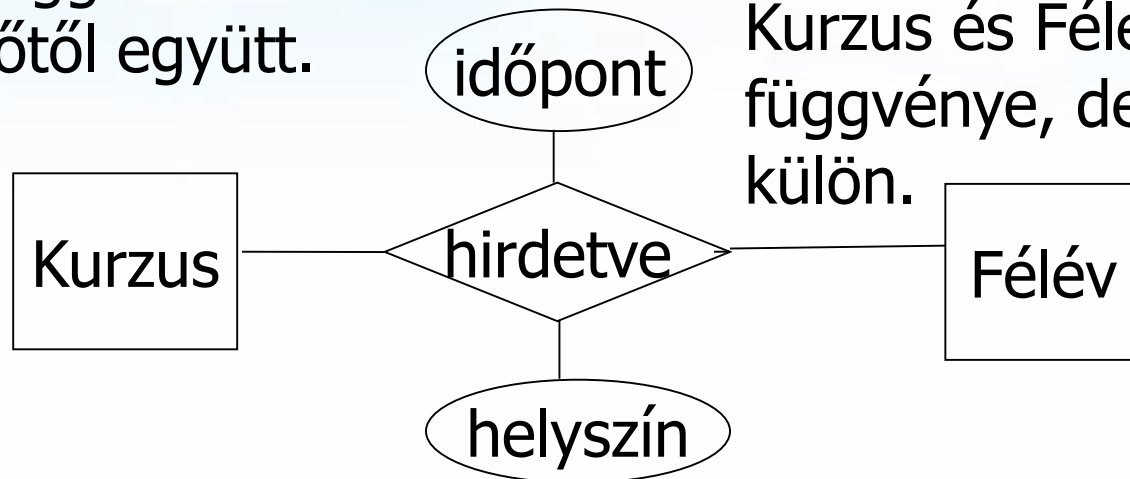
ami körül körben van az öt egyedhalmaz: VEVŐ, TERMÉK, CSATORNA, IDŐ, KEDVEZMÉNY

Kapcsolatnak is lehet attribútuma

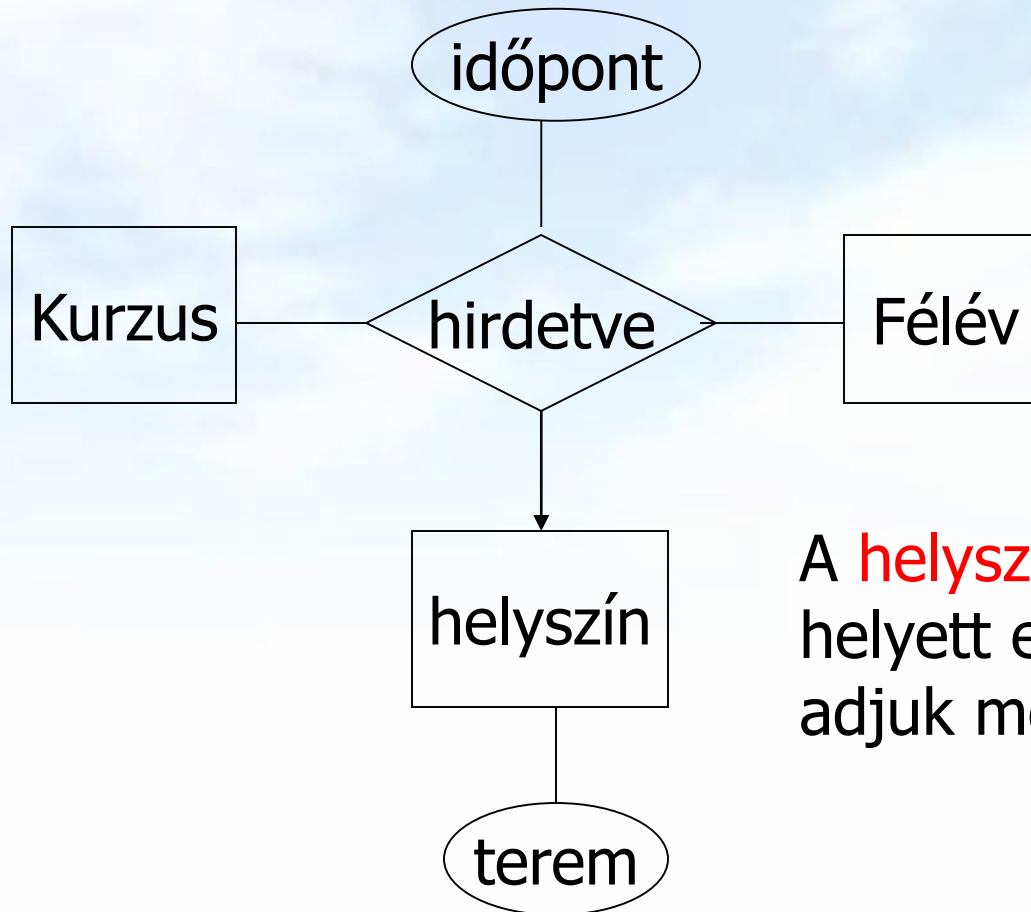


A sör **ára** függ a Sörtől és a Sörözőtől együtt.

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

Mit tanultunk eddig?

A mai előadáson eddig áttekintettük (Tk.2.1.-2.2; Tk 4.1.)

- Relációs adatmodell (Tk.2.1-2.2.)
 - Mit jelent a reláció sémája és a reláció előfordulása?
- E/K diagram alapelemei (Tk.4.1.)
 - Mi az egyed, tulajdonság (attribútum), hogyan jelöljük?
 - Mi a kapcsolat, hogyan jelöljük a kapcsolatok típusait?
 - Sokágú kapcsolatok átalakítása binárisra példák

Most jön: E/K-diagram átírása relációsémákra (Tk 4.5.)

- Egyedhalmazok átírása relációkká
- Kapcsolatok átírása relációkká
- Egyszerűsítés, összevonások

Ezután folytatjuk E/K-diagram (Tk 4.2.-4.4.és 4.6.)

- Gyenge egyedhalmazok kezelése
- Osztályhierarchia átalakítása relációkká

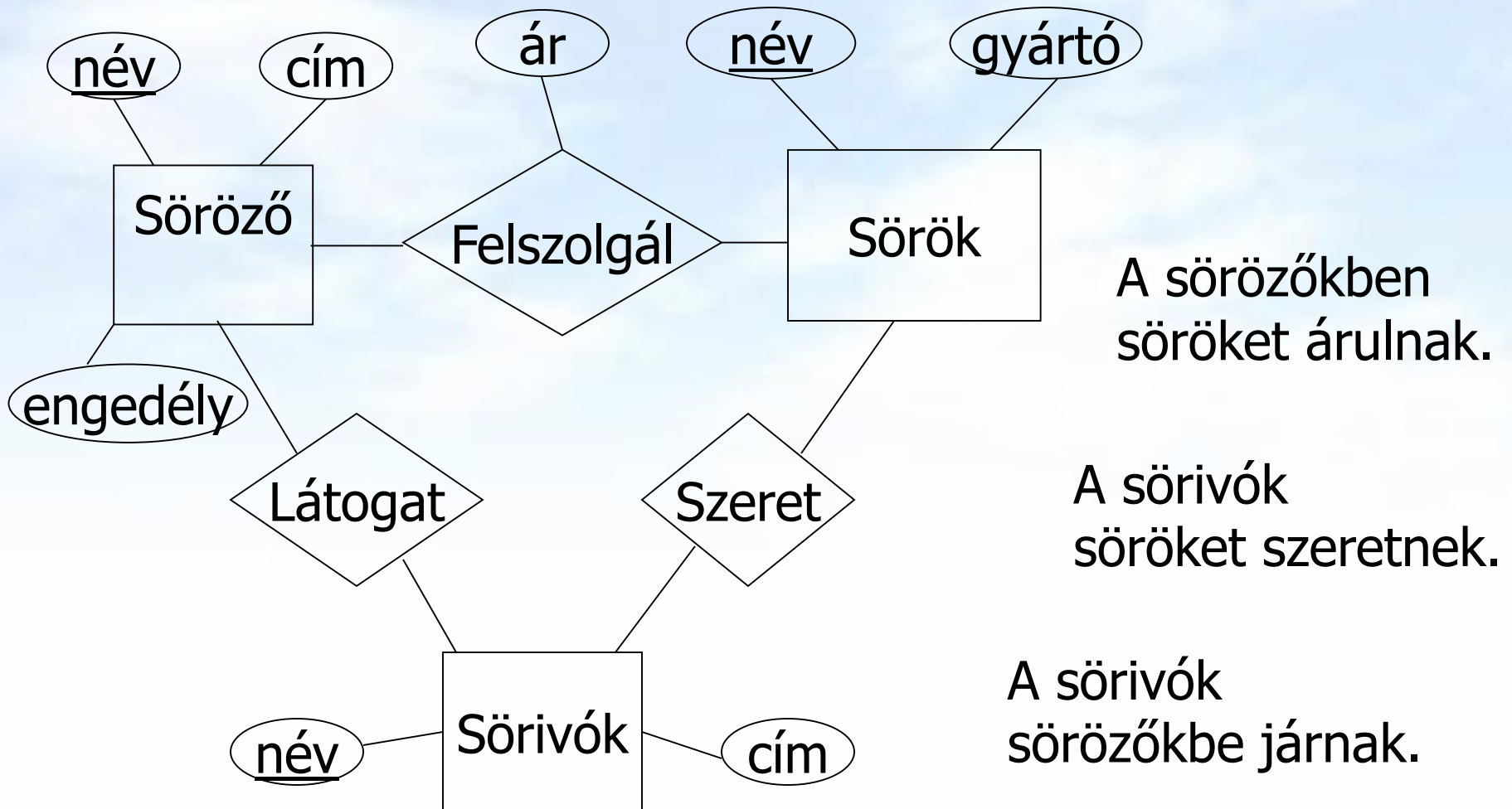
E/K diagram átírása relációs adatbázisra

Mi minek felel meg:

- egyedhalmaz séma $E(A_1, \dots, A_n)$ ↔ relációséma $E(A_1, \dots, A_n)$
- 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 E_i egyedhalmaz, A_j saját tulajdonság ↔ relációséma, ahol K_i az E_i (szuper)kulcsa
- E/K modell ↔ Relációs adatmodell

Példa: E/K diagram átírása relációkká ---1

Az egyedek és sok-sok kapcsolatok átírása



Példa: E/K diagram átírása relációkká ---2

Az egyedek és sok-sok kapcsolatok átírása

- Az egyedhalmazok átírása
(aláhúzás jelöli a kulcs attribútumokat)

Sörök(név, gyártó)

Sörözők(név, cím, engedély)

Sörivők(név, cím)

- Sok-sok kapcsolatok átírása

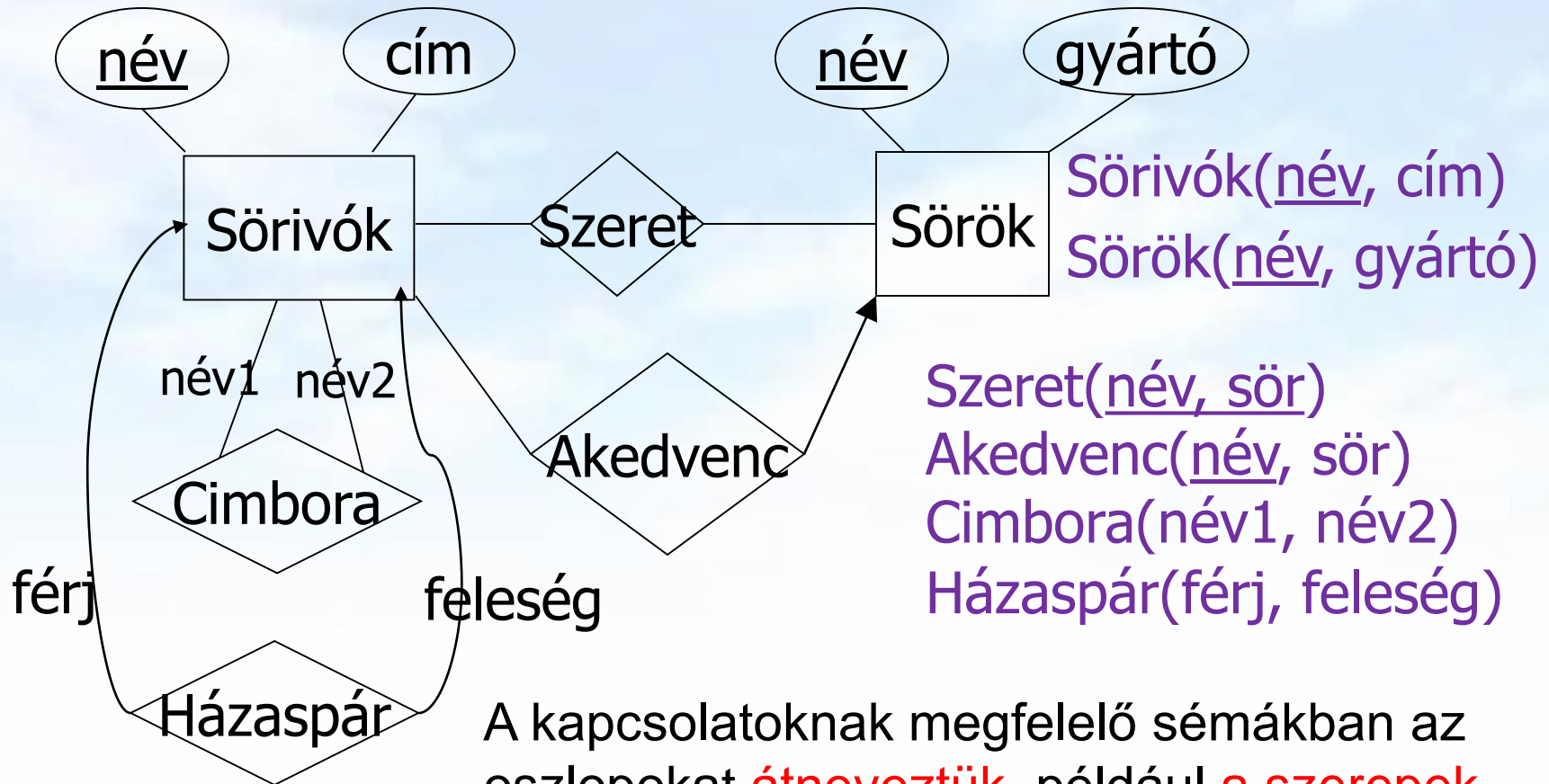
Szeret(név, sör)

Felhasználó(söröző, sör, ár)

Látogat(név, söröző)

Példa: E/K diagram átírása relációkká ---3

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 ---4

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

Tervezési témakört folytatjuk

Az E/K diagram többi elemével folytatjuk:

- Alosztályok és öröklődés
(speciális „is-a”/„az-egy” kapcsolat)
- Kulcsok és további megszorítások
- Gyenge egyedhalmazok kezelése

majd ezek relációsémákra való átalakítása jön:

- E/K diagram átalakítása relációsémákra
- Osztályhierarchia átalakítása relációkká

Alosztályok és öröklődés

Speciális „is-a” / „az-egy” kapcsolat

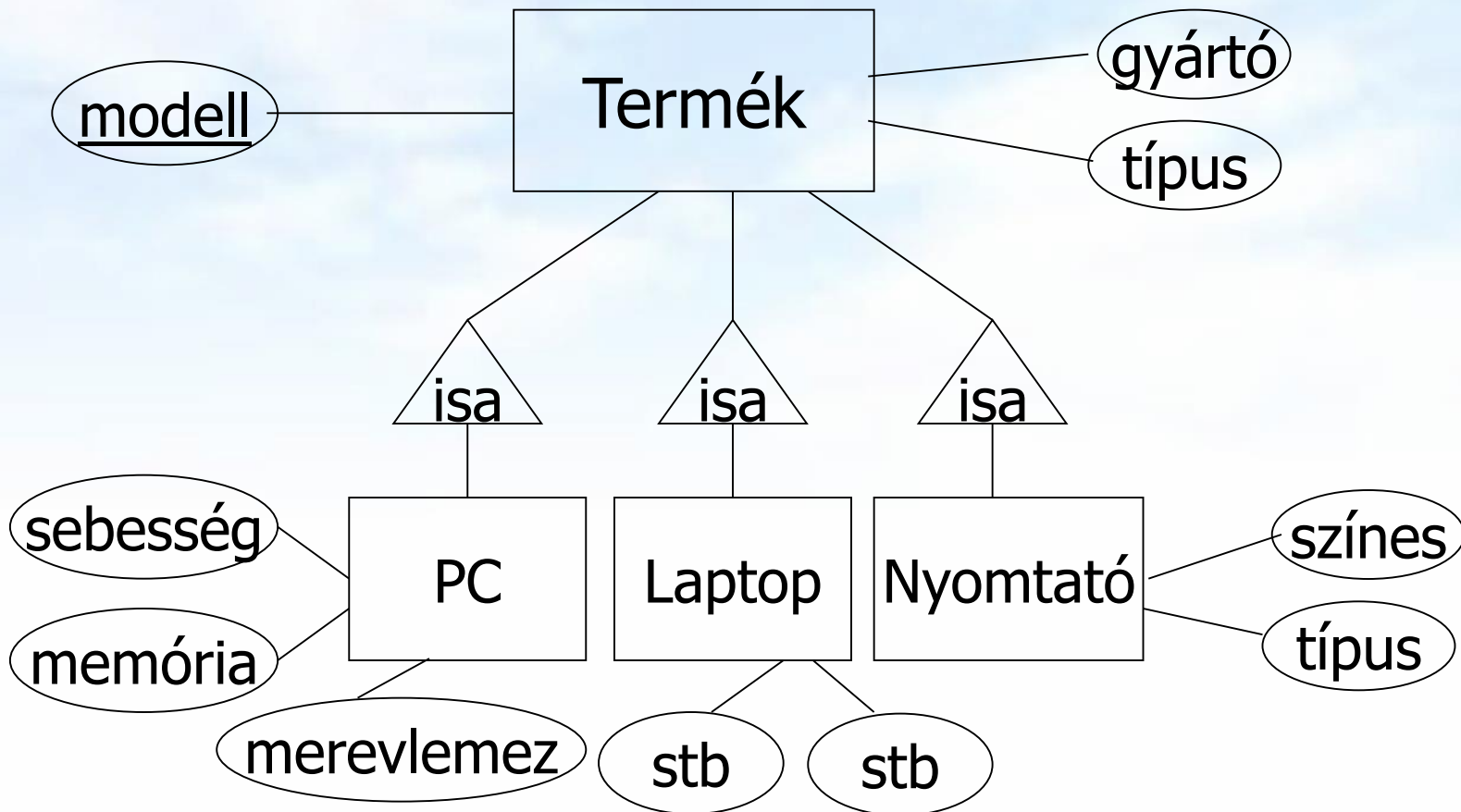


Alosztály - speciális eset, kevesebb egyed előfordulás, több tulajdonság (attr./kapcs.)

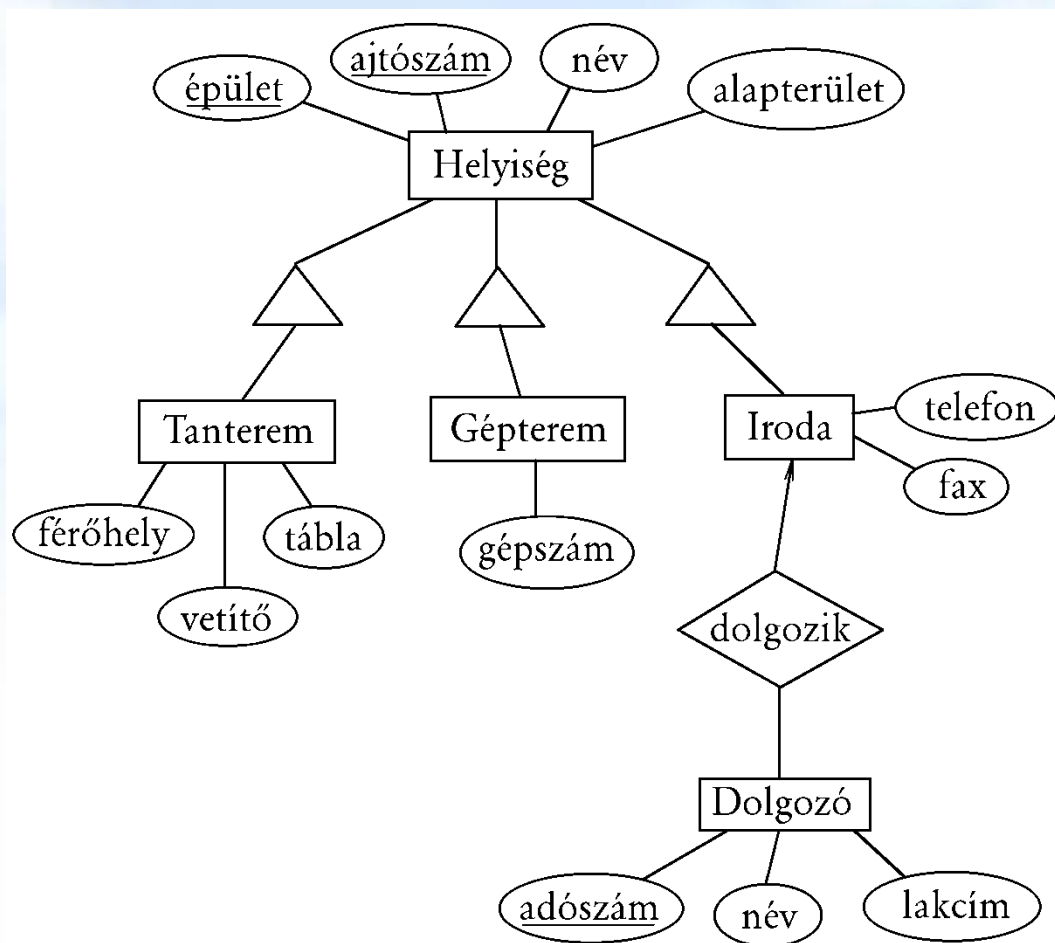
Az alosztály kapcsolatot „az egy” háromszögek jelölik (ahol az őszosztályra mutat a háromszög felső csúcsa)

Az alosztályok rendszere fát alkot (nincs többsz.öröklés)

Példa: „is-a” / „az-egy” kapcsolatra Termékek (Tankönyv feladata)

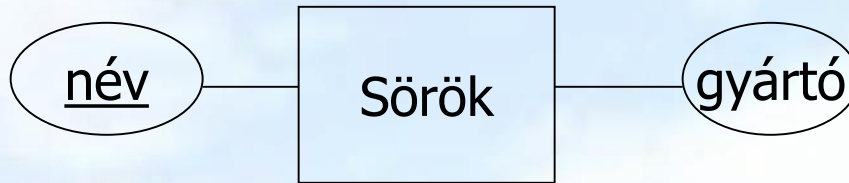


Másik példa: „is-a” (az-egy) kapcsolatra

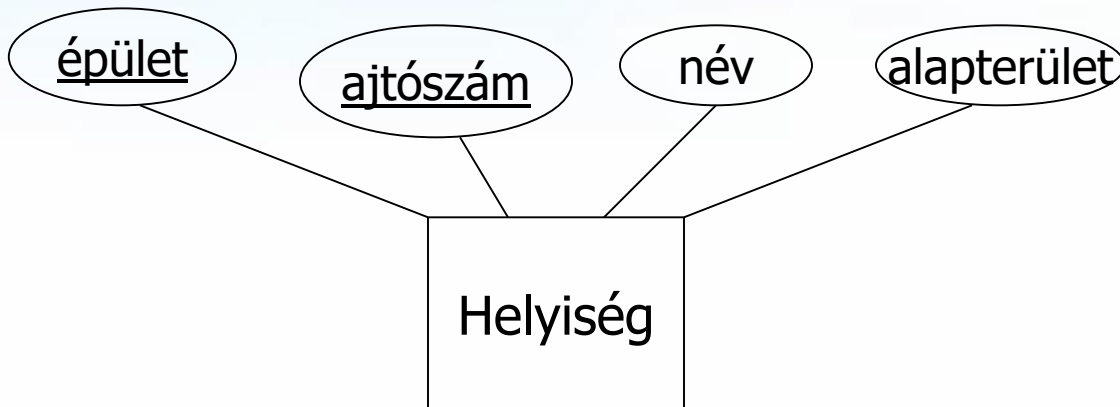


Kulcs megszorítás jele: aláhúzás

Példa egyszerű kulcsra: név a Sörök elsődleges kulcsa:



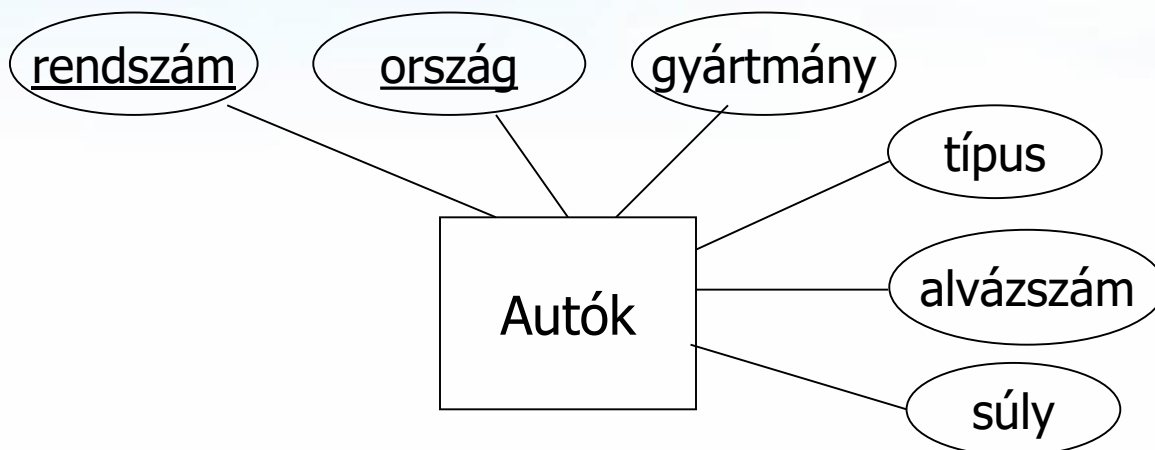
Példa összetett kulcsra: épület, ajtószám két-attribútumos elsődleges kulcsa a Helyiség-nek:



Kulcs (minimális szuperkulcs)

Példa kulcsra: rendszer, ország

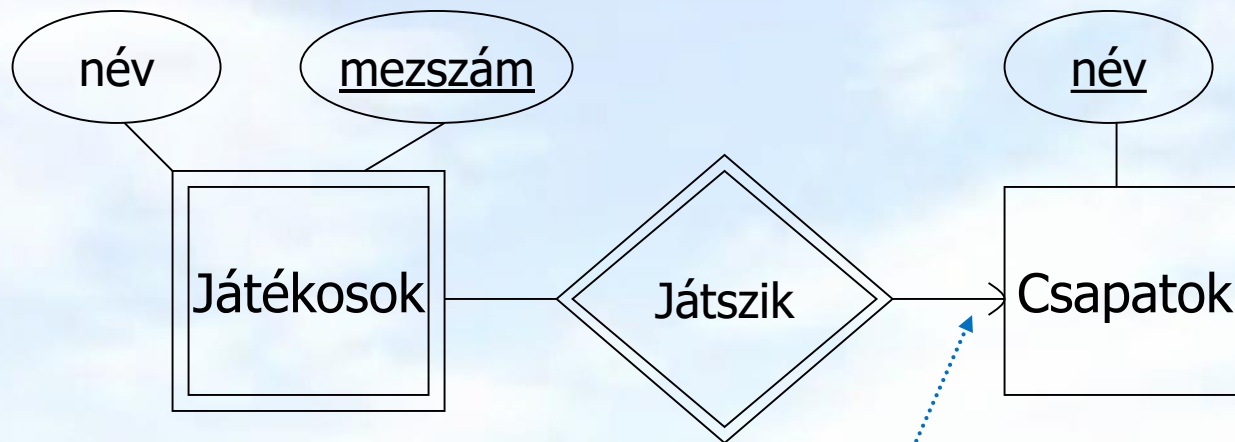
Autók egyedhalmazt leíró tulajdonságok közül több kulcs is van, de egy elsődleges kulcsot kiválasztunk és csak azt jelöljük aláhúzással. Például egyszerű kulcs az alvászám (ez is egyértelműen azonosítja), vagy összetett kulcs rendszer, ország (minimális, vagyis nincs valódi részhalmaza, ami azonosítaná).



Hivatkozási épség megszorítás jele a kerek végződés —)



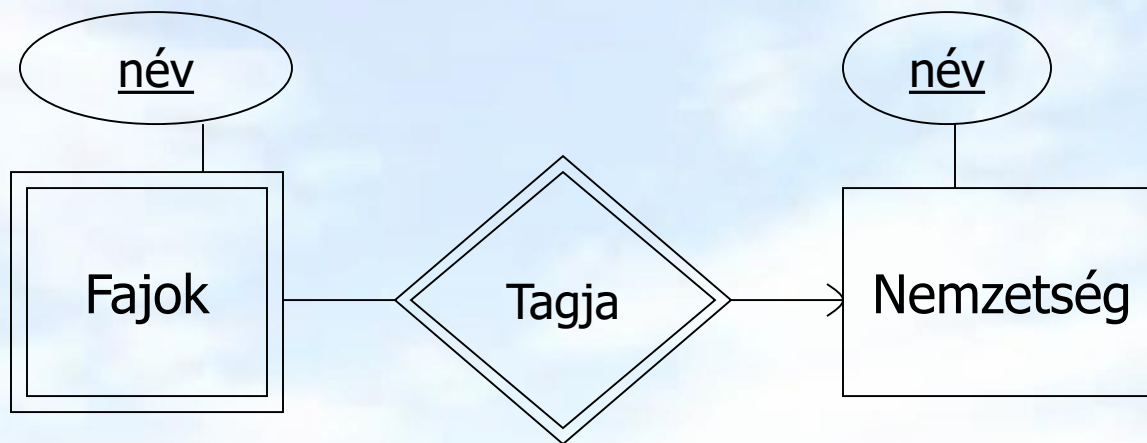
Erős és gyenge egyedhalmaz



- Dupla téglalappal jelöljük a **gyenge** egyedhalmazt.
- Dupla rombusz jelöli a támogató sok-egy gyenge kapcsolatot.

A **támogató** kapcsolatoknak kerek nyílban kell végződniük az egy oldalon: A kerek végződés jelzi, hogy minden játékoshoz kötelezően tartozik egy csapat, amely az azonosításhoz használható.

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

- **valóságghű modellezés:**
 - megfelelő tulajdonságok tartozzanak az egyedhalmazhoz, 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:**
 - Ne használjunk egyedhalmazt, ha egy attribútum éppúgy megfelelne a célnak, például a **vizsgajegy** egyedhalmaz helyett **jegy** tulajdonságot használjunk.

E/K-diagram átírása relációkra

Gyenge egyedhalmaz átírása



Hostgépek(hostNév, cím)

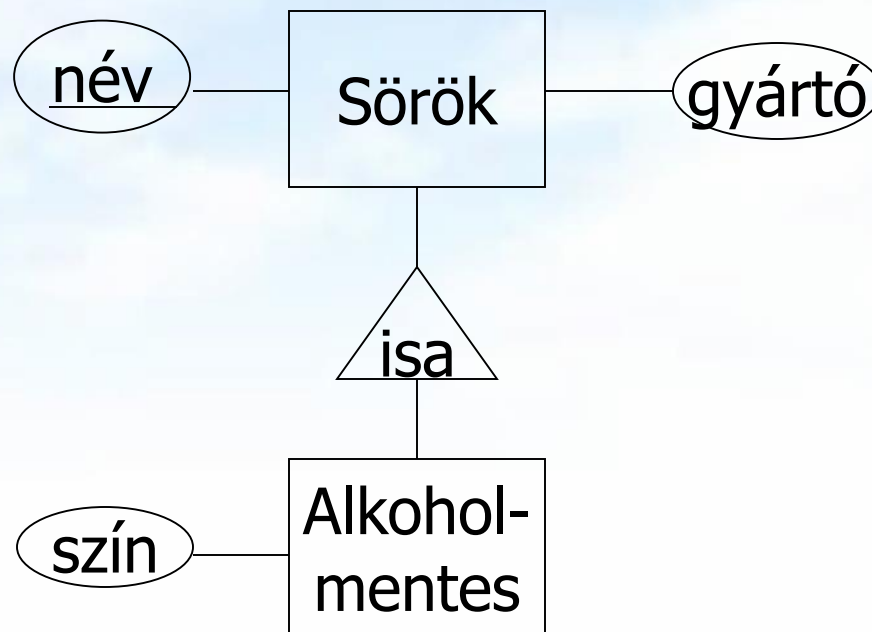
Logins(loginNév, hostNév, jelszó)

~~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, mint például, hogy egy adott gyártó milyen söröket gyárt (egy táblából jön az infó, máskülönben össze kellene kapcsolni a két táblát).

Objektumorientált megközelítés ---2

név	gyártó
Bud	Anheuser-Busch

Sörök

név	gyártó	szín
Summerbrew	Pete's	világos

Alkoholmentes

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

Nullértékek használatával ---3

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

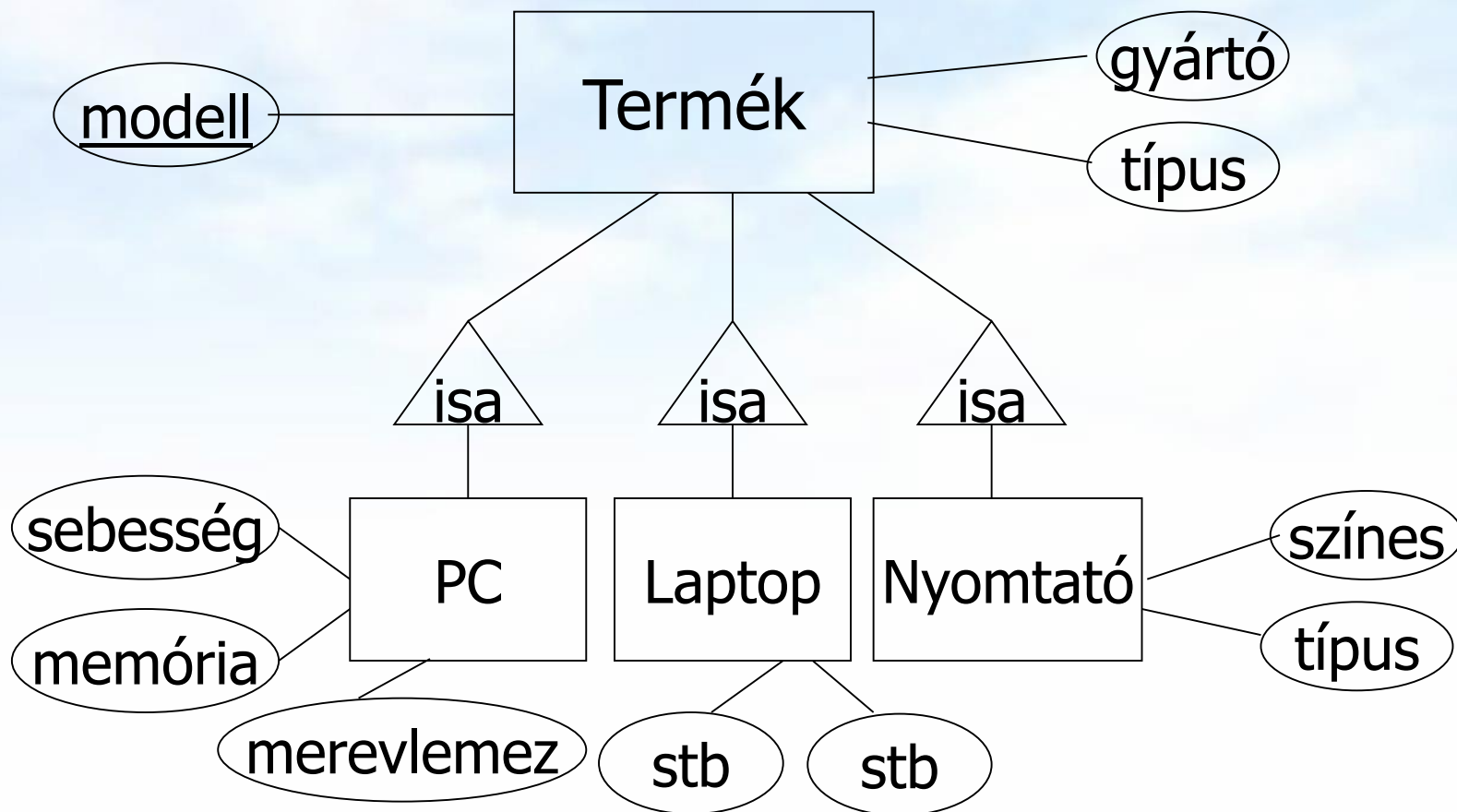
Sörök

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.

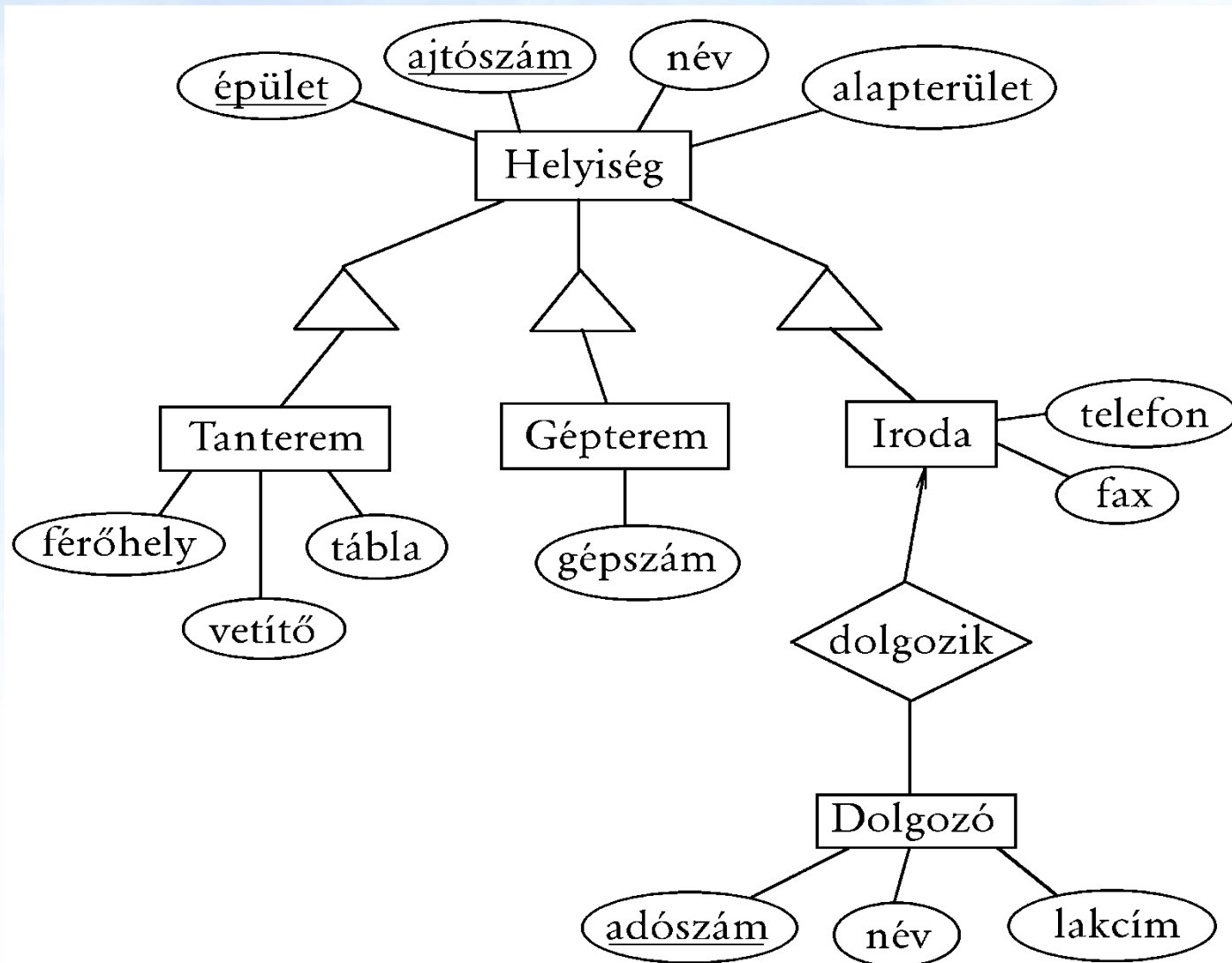
Általában kevesebb hely elég a tárolásra, kivéve ha nagyon sok attribútum marad nullértékű.

Példa: Termékek (Tankönyv feladata)

H.F.: Az alábbi E/K diagramot írjuk át mind a három megközelítéssel relációsémákra! Hasonlítsuk össze!



Feladat: Alosztály átírása relációkká



Modellezési feladatok gyakorlatra

- **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 gyakorlatra

- **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 gyakorlatra

- **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.

Oracle-gyakorlat: HR séma táblái

Dolgozók, Osztály, Fiz_kategória

EMPLOYEES

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALA
100	Steven	King	SKING	515.123.4567	17-JUN-87	AD_PRES	240
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	170
102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	170
103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT_PROG	90
104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-91	IT_PROG	60
107	Diana	Lorentz	DLORENTZ	590.423.5567	07-FEB-99	IT_PROG	42
124	Kevin	Mourgos	KMOURGOS	650.123.5234	16-NOV-99	ST_MAN	58
141	Trenna	Rajs	TRAJS	650.121.8009	17-OCT-95	ST_CLERK	35
142	Curtis	Davies	CDAVIES	650.121.2994	29-JAN-97	ST_CLERK	31
144	Charles	Johnson	CJOHNSO	650.121.2004	15-MAR-98	ST_CLERK	26
145	Shelley	Greenberg	SGREENB	650.121.2004	09-JUL-98	ST_CLERK	25
146	Dena	Raphaely	DRAPHAEL	650.121.2004	22-JAN-99	ST_MAN	40

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700

DEPARTMENTS

GRA	LOWEST_SAL	HIGHEST_SAL
A	1000	2999
B	3000	5999
C	6000	9999
D	10000	14999
E	15000	24999
F	25000	40000

JOB_GRADES

Kérdés / Válasz

- **Köszönöm a figyelmet! Kérdés/Válasz?**
- **Oracle gyakorlaton:** Az Oracle adatbázisok elérése, sqldeveloper kliens, csatlakozás az adatbázishoz.
- A táblák előkészítése, táblák létrehozása: $R(A_1, \dots, A_n)$, sémaleírásban az A attribútumok típusának megadása (Oracle implementációban) **milyen standard típusok** közül választhatunk: mi a különbség a CHAR és a VARCHAR között? Megszorításoka megadása később.
- A táblákat a létrehozásuk után feltöltjük adatsorokkal
- **Következő alkalommal:** Lekérdezések SQL SELECT