

1.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

Bevezetés/Előkészítés:

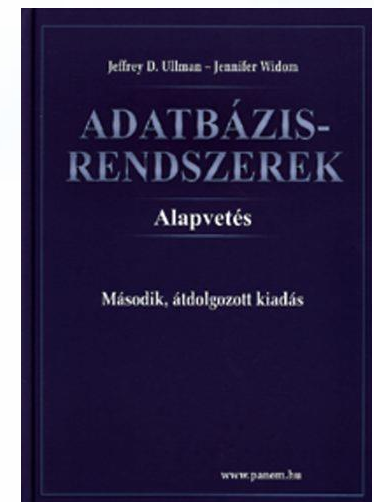
Adatmodellezésre feladatok, E/K modell, E/K diagram leképezése relációsémákra

A mai témakörök a Tankönyvben:

2.1.-2.2. Adatmodellek, relációs modell

4.1. E/K-modell elemei (folyt. 4.előadás)

4.5. 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)

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

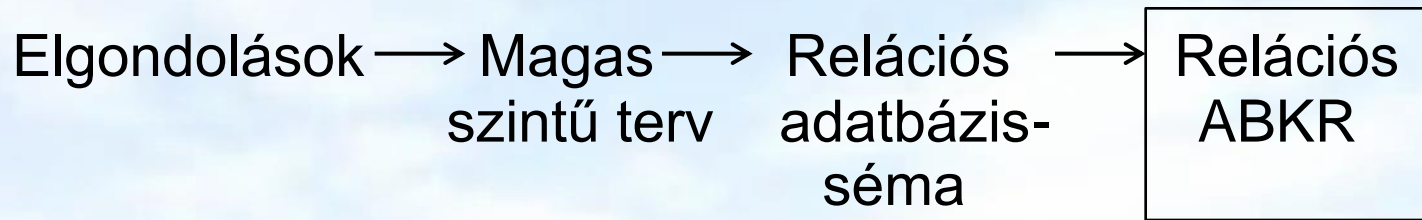
Példa 4

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

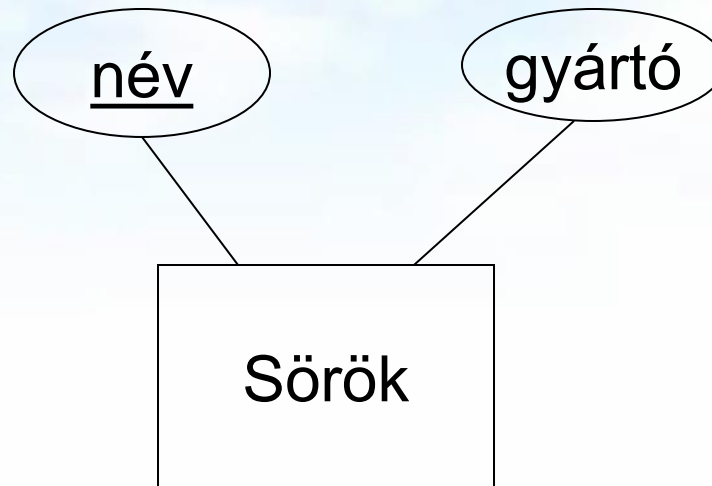
- **Egyed-kapcsolat modell: E/K modell**
(Entity-relationship ER) alapfogalmak:
- **Egyedhalmazok** (absztrakt objektumok osztálya)
 - Miről gyűjtünk adatokat?
 - Mit tegyünk egy gyűjteménybe? - hasonlóság
 - Hasonló egyedek összessége
- **Attribútumok**
 - Megfigyelhető tulajdonságok, megfigyelt értékek
 - 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.

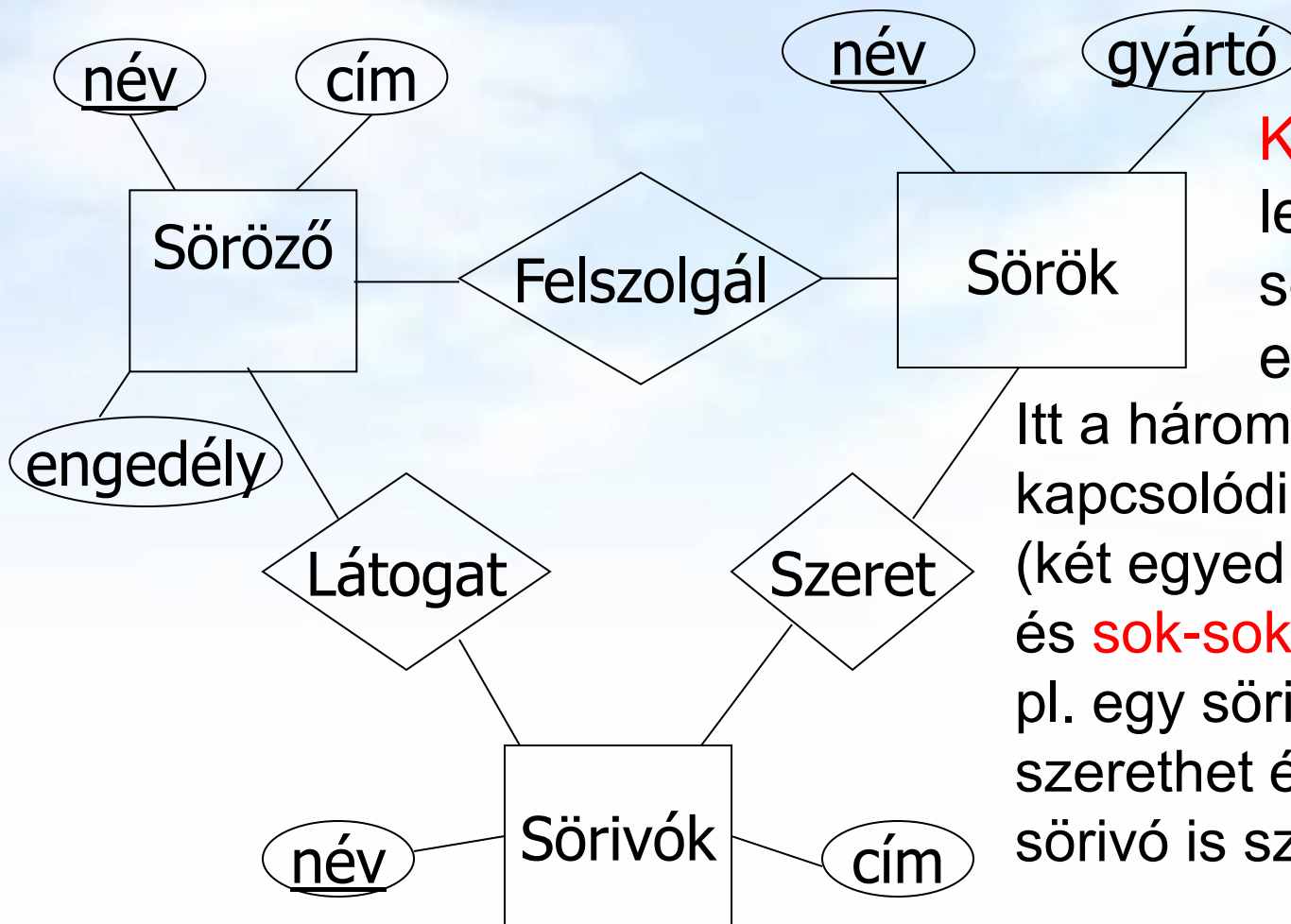
E/K-diagram: Kapcsolatok

- 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)

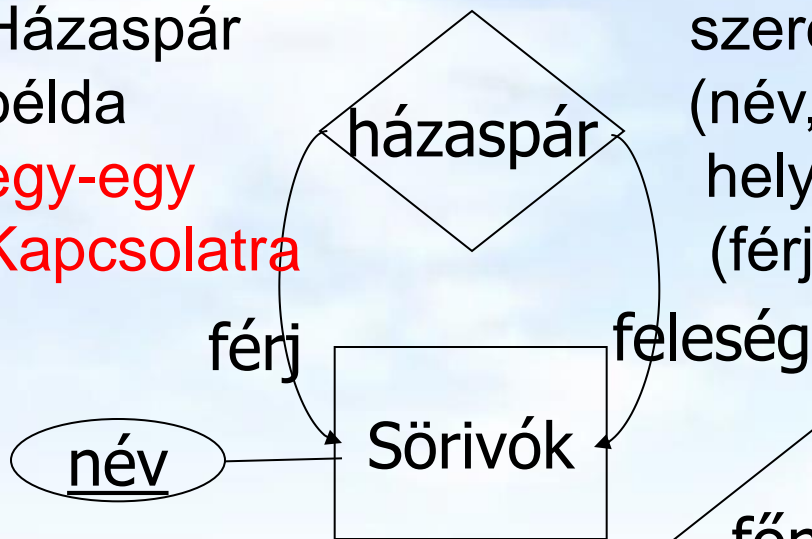


sok-egy kapcsolat:
(nyíl) K előfordulásaiban minden E1-beli egyedhez legfeljebb egy E2-beli egyed tartozhat!

Egy egyedhalmaz önmagával is kapcsolódhat: Szerepek (Roles)

Házaspár
példa

egy-egy
Kapcsolatra



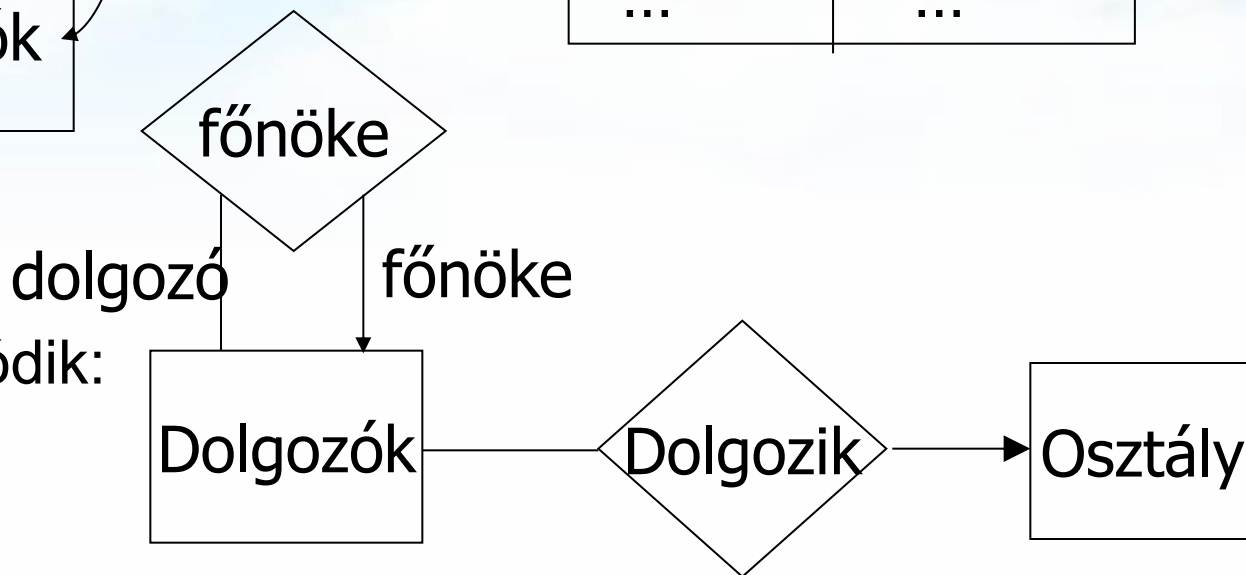
szerepkör
(név,név) pár
helyett lesz
(férj,feleség)

Kapcsolat előfordulás

férj	feleség
Bob	Ann
Joe	Sue
...	...

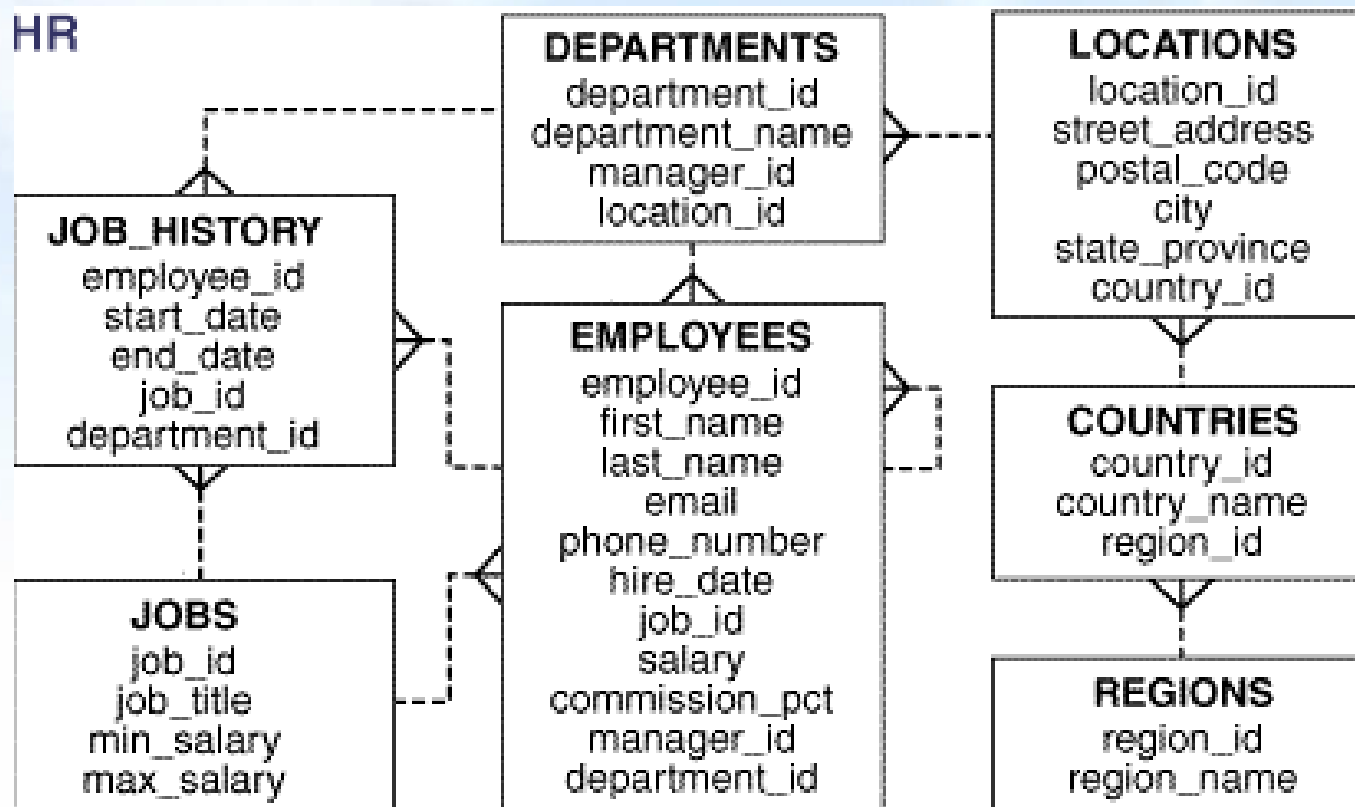
Egy másik példa,
ahol egyedhalmaz
önmagával kapcsolódik:
dolgozó - főnöke

sok-egy kapcsolat

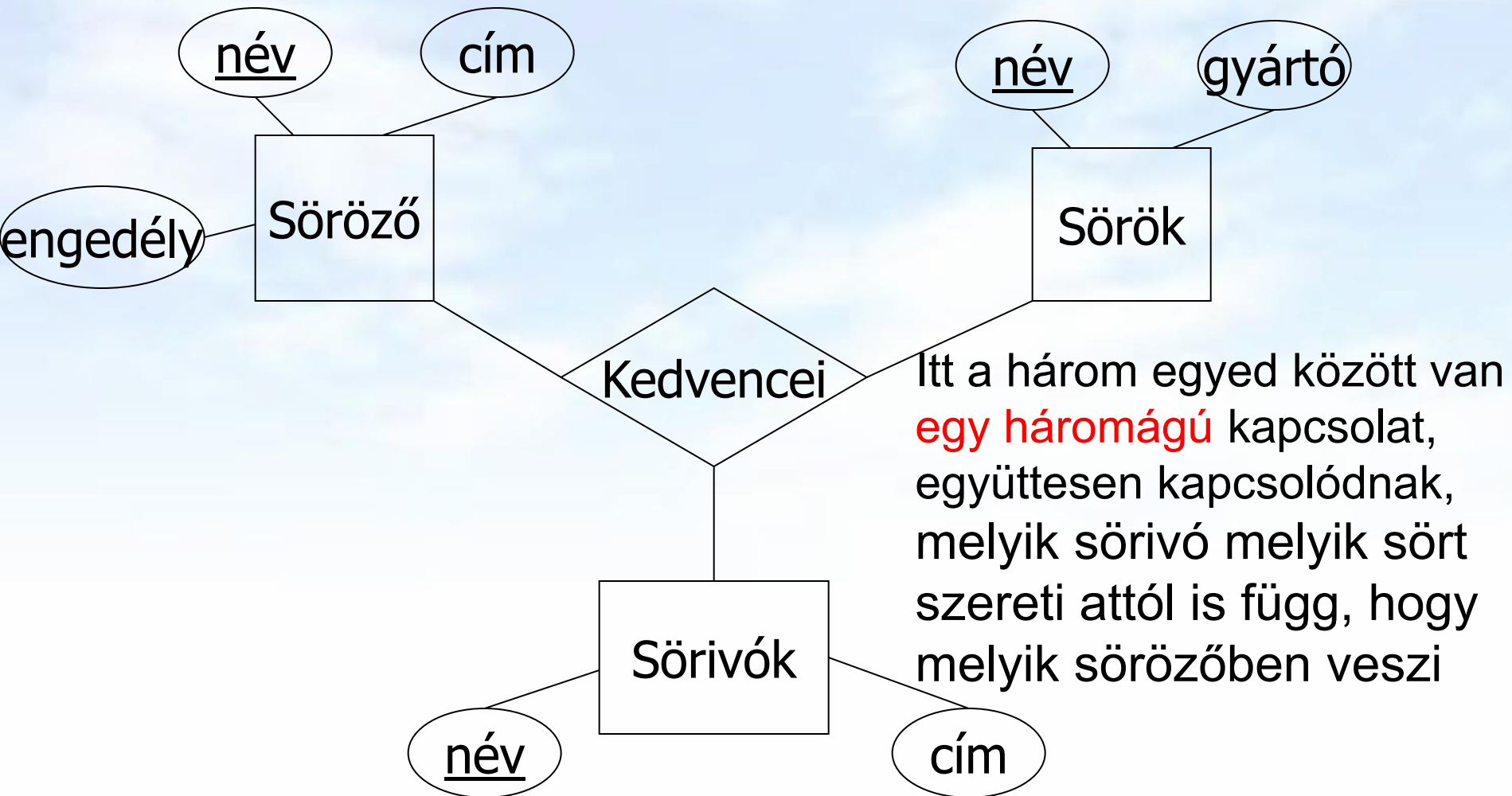


Példa a gyak-on: Oracle HR séma táblái

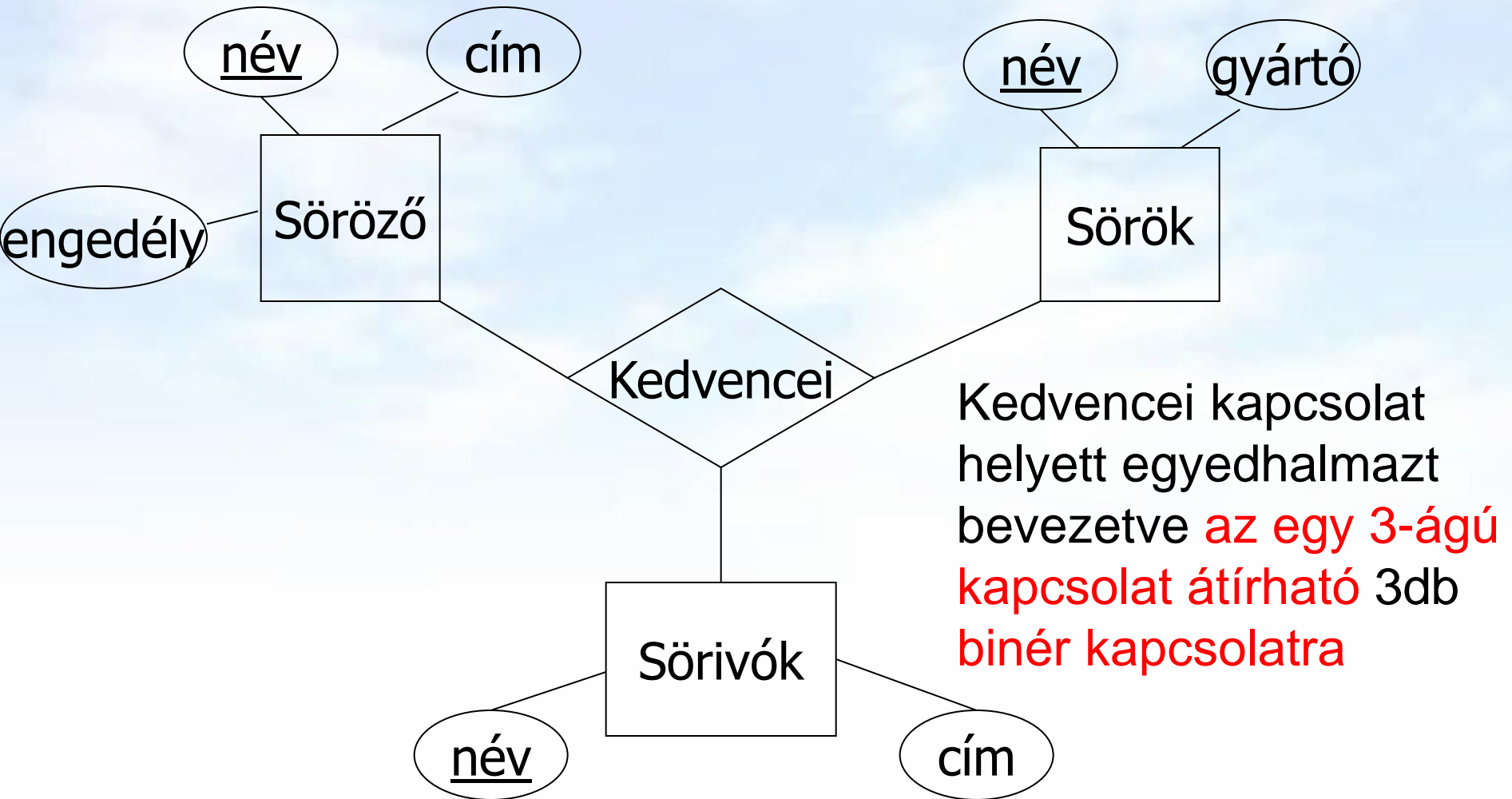
Itt új jelöléssel csak bináris kapcsolatok lehetnek!



Példa: Többágú (3-ágú) kapcsolatra



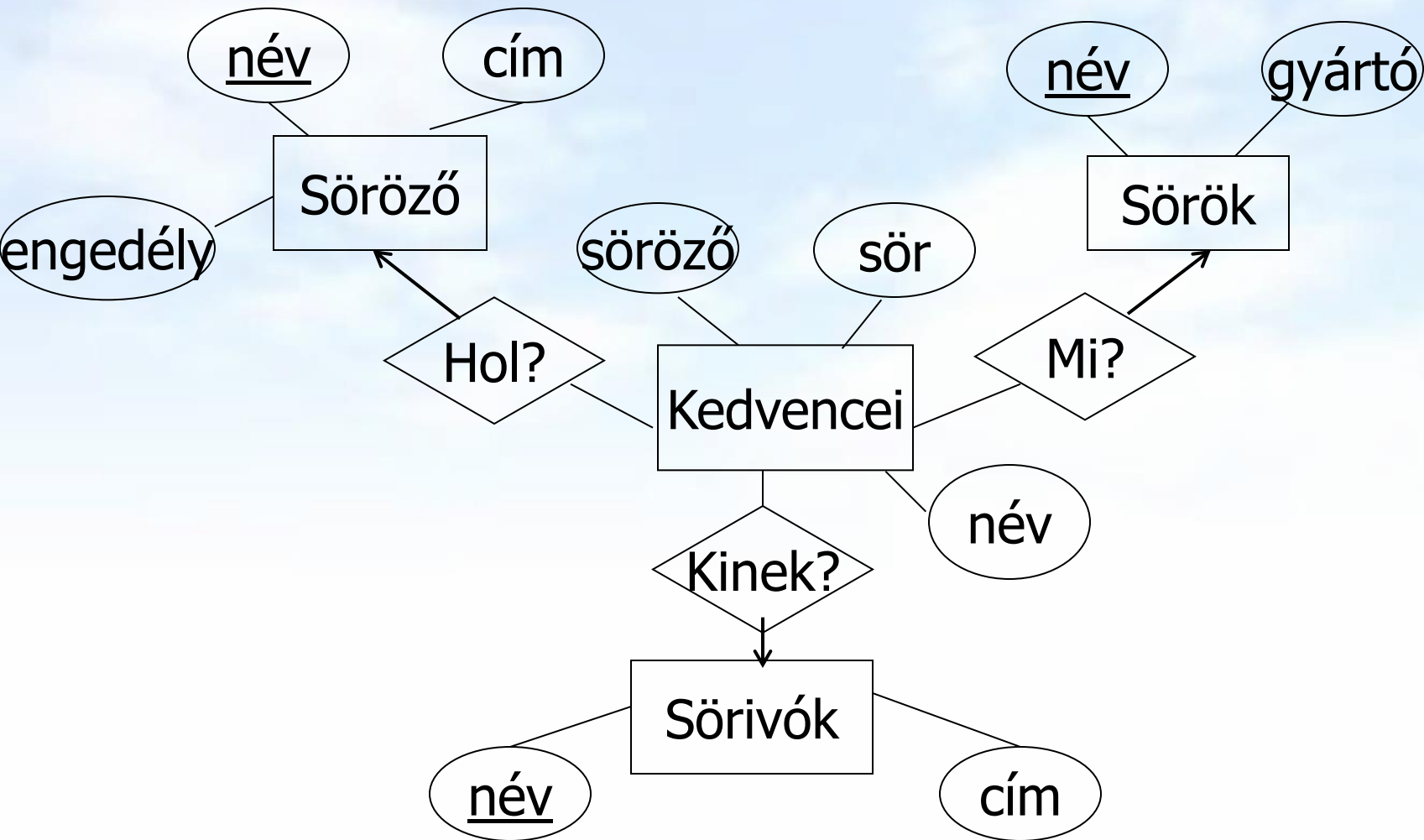
Példa: Átírható-e bináris kapcsolatra?



Példa: Többágú kapcsolat átírása bináris kapcsolatokra

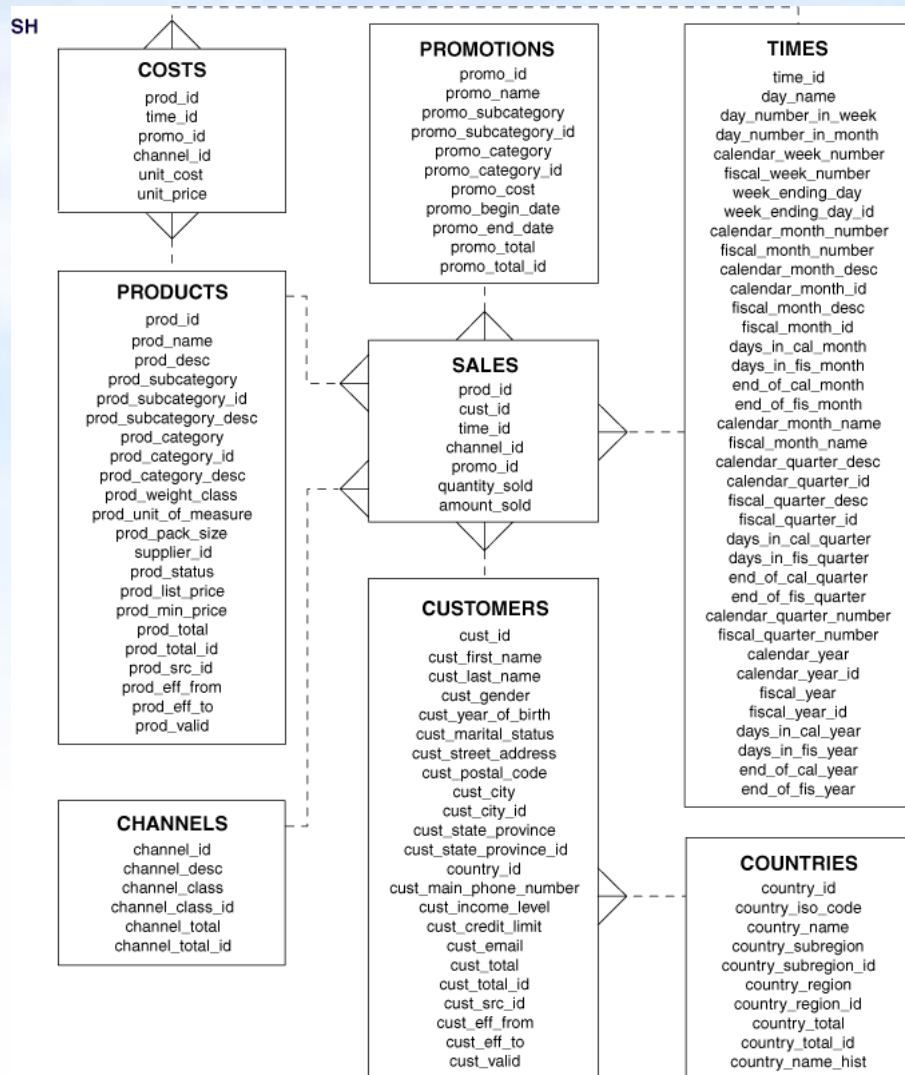
- Kedvencei kapcsolat helyett egyedhalmazt bevezetve ez **a 3-ágú kapcsolat átírható 3db binér kapcsolatra**
- Vagyis Kedvencei kapcsolat helyett egy új egyedhalmazt vezetünk be, aminek három attribútuma van: a név, söröző, sörivó
- és ezt az új egyedhalmazt középen téglalappal jelöljük, és ez kapcsolódik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

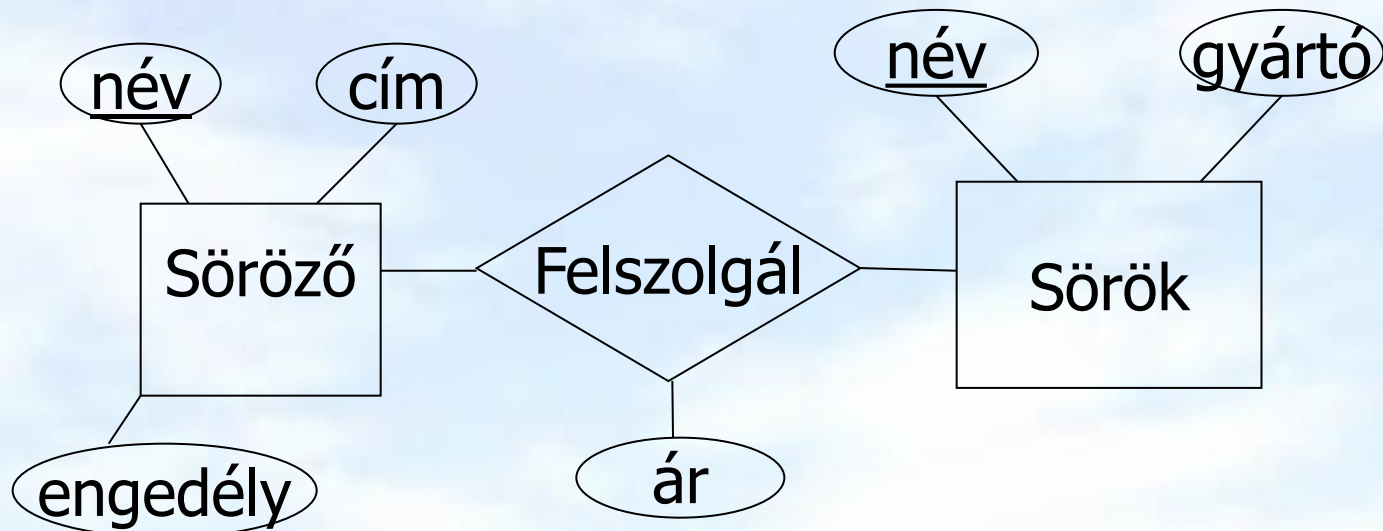


Többágú kapcsolat lett itt is feloldva, mint előbb!
E/K jelölésben középen 5-ágú kapcsolat lenne a SALES vagyis ELADÁS



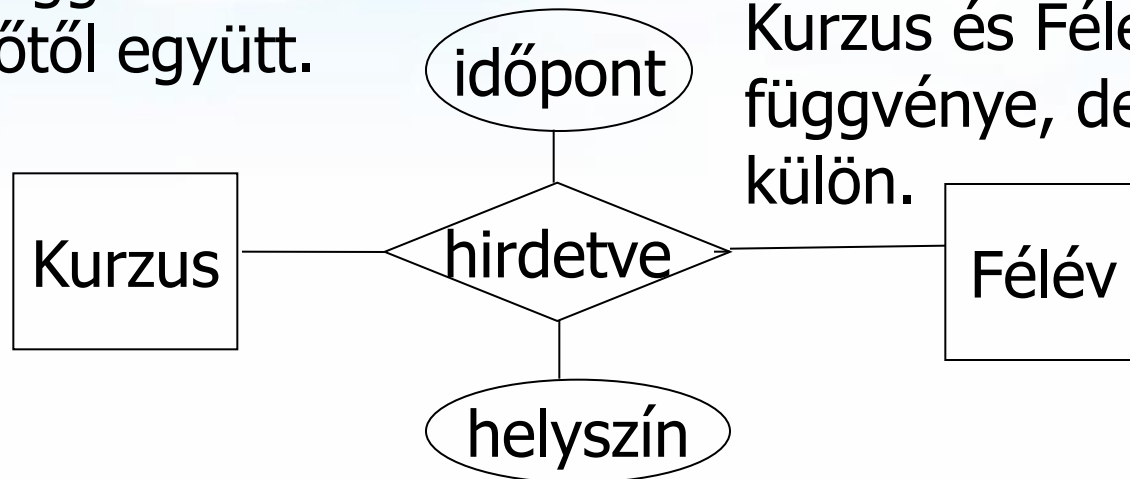
ami körül körben van az öt egyedhalmaz: VEVŐ, TERMÉK, CSATORNA, IDŐ, KEDVEZMÉNY

Kapcsolatnak is lehet attribútuma

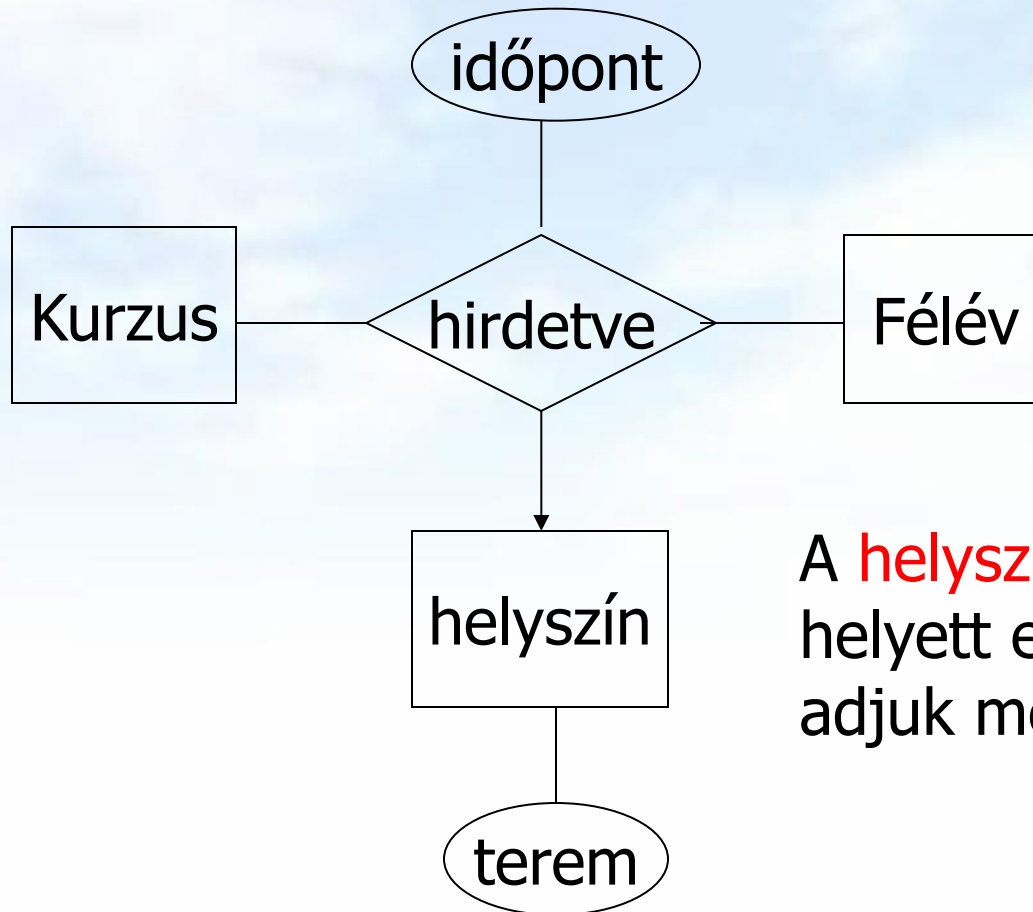


A sör **ára** függ a Sörtől és a Sörözőtől együtt.

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

E/K-diagram átírása relációkra

Előtte: Relációs adatmodell (Tankönyv 2.1-2.2.)

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

- Egyedhalmazok átírása relációkká
- E/K-kapcsolatok átírása relációkká
- Egyszerűsítés, összevonások

Az átírást is a 4.előadáson folytatjuk (Tk 4.6.)

- Gyenge egyedhalmazok kezelése
- Osztályhierarchia átalakítása relációkká

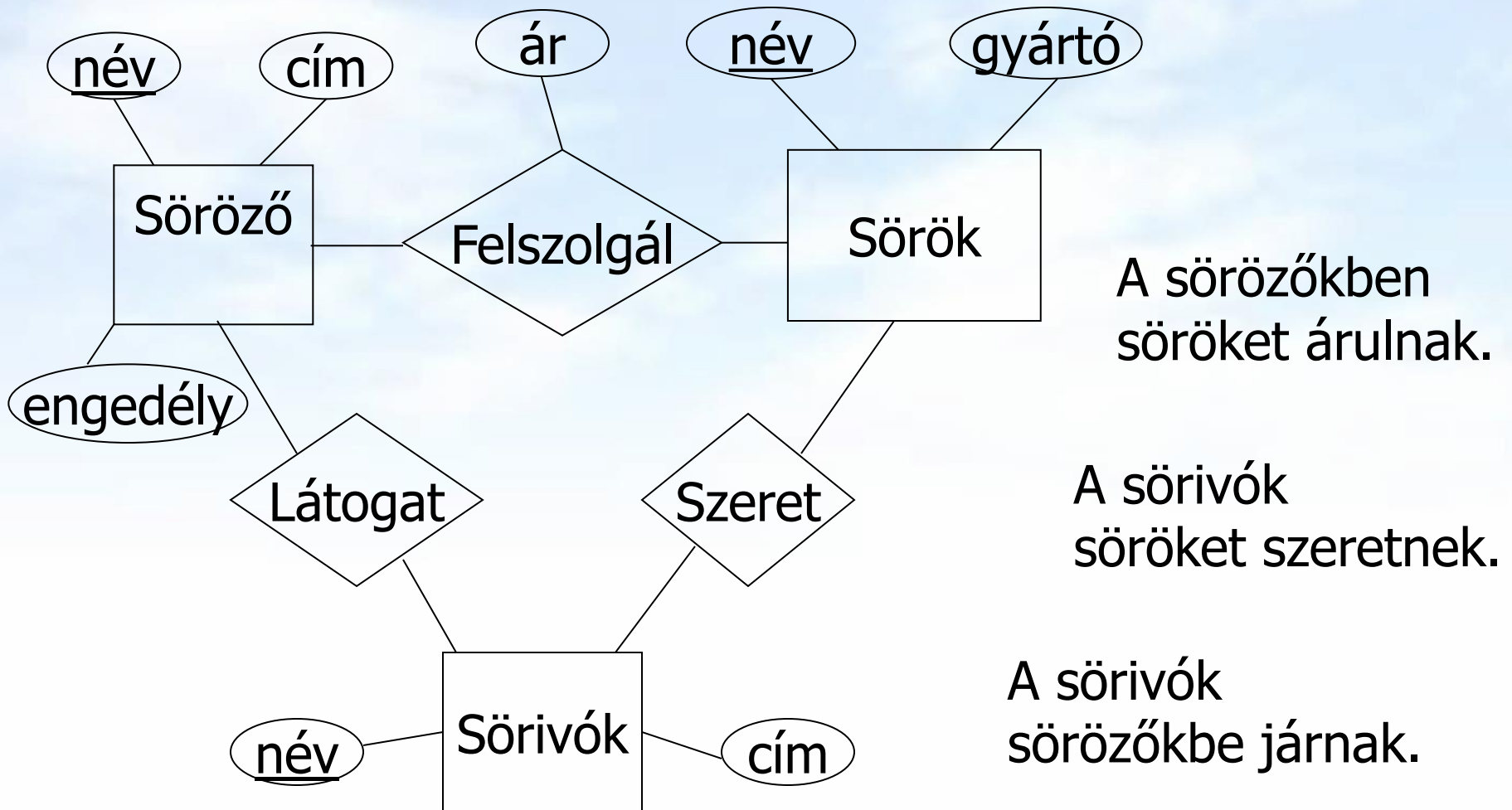
E/K diagram átírása relációs adatbázistervre

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, város, tulaj, engedély)

Sörivők(név, város, tel)

- Sok-sok kapcsolatok átírása

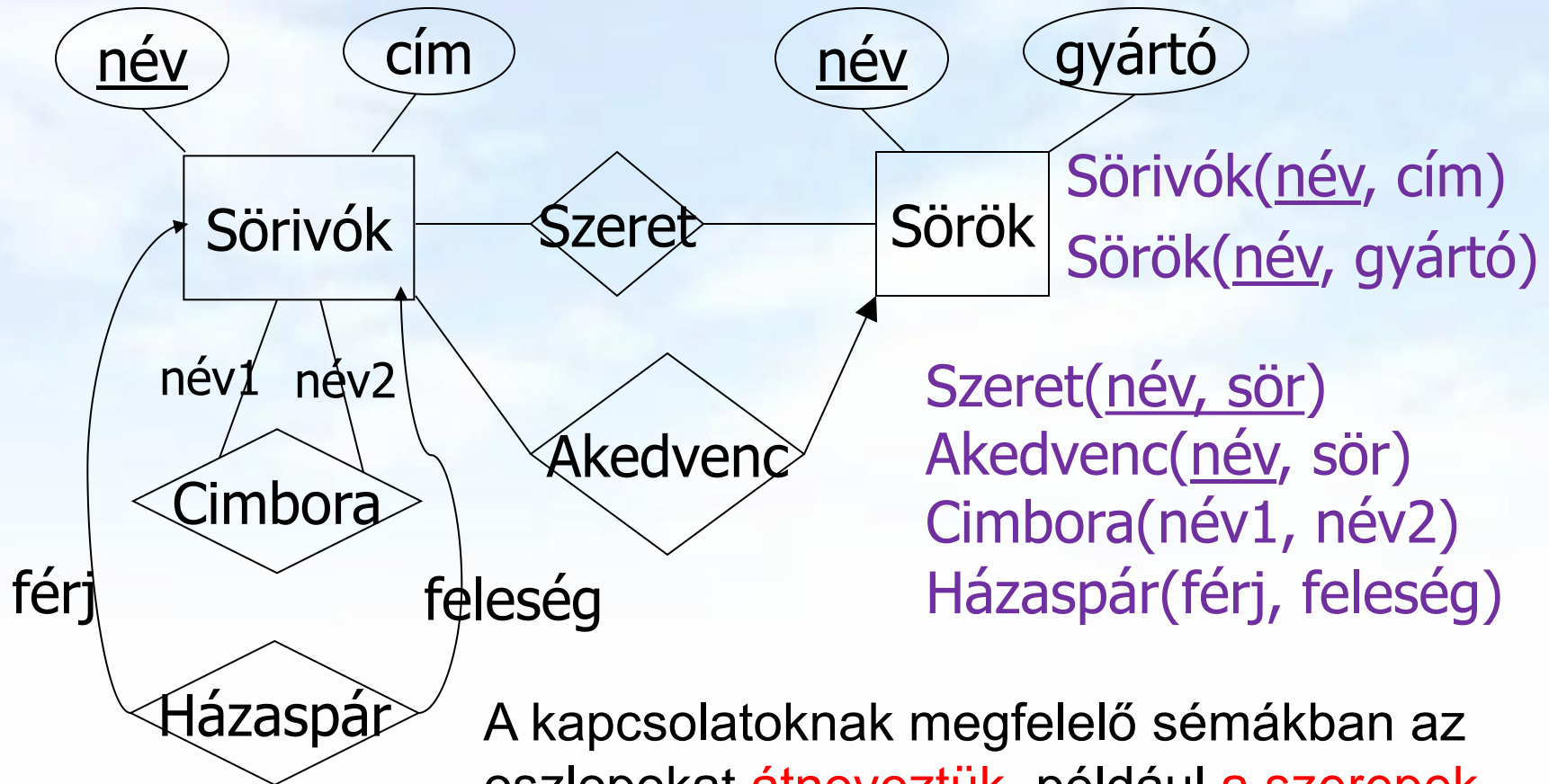
Szeret(név, sör)

Felhasználó(söröző, sör, ár)

Látogat(név, söröző)

Példa: E/K diagram átírása relációkká ---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

Ezek voltak a tervezési alapok:

Mit tanultunk ma?

- Mit jelent a reláció sémája és a reláció előfordulása?
- Mi a kapcsolat, hogyan jelöljük, kapcsolat típusait?
- Sokágú kapcsolatok átalakítása binárisra példák
- E/K diagram relációs sémákra való átalakítása példák

Tervezési témakört folytatjuk a 4.előadáson

- Kulcsok és további megszorítások
- Alosztályok és öröklődés, spec „is-a” (az-egy) kapcsolat
- Gyenge egyedhalmazok

Folytatás: SQL1.pdf

- Ezt az 1.előadást folytatjuk a 2.résszel:
Hogyan hozzuk létre a táblákat SQL-ben, hogyan készítjük elő a lekérdezésekhez, lásd [SQL1.pdf](#)
- Ezután köv 2.előadáson: Lekérdezések kifejezése relációs algebrában és SQL SELECT utasítással.
- **1.gyakorlaton:** Adatmodellezésre feladatok, példák E/K diagram felrajzolása és átalakítása relációs modellre. A gyakorlatra szóló modellezési feladatok megtalálhatóak itt a TERV1.pdf diasorozat végén. A gyakorlatokon megbeszéljük a demo adattáblák kialakításához milyen E/K modell áll a háttérében, lásd Oracle HR séma Dolgozó és Osztály tábláit.

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.