

ADATMODELLEK

Adatbázisok 1

Dóka-Molnár Andrea

andrea.molnar@math.ubbcluj.ro



BABEŞ-BOLYAI TUDOMÁNYEGYETEM
Matematika és Informatika Kar



Áttekintés/Tematika

1. Adatbázis-kezelő rendszerek általános jellemzői.
2. **Adatmodellezés**
 - **Egyed/kapcsolat adatmodell**
 - **Relációs adatmodell**
 - **Az E/K diagram átalakítása relációs adatmodellé**
3. A relációs algebra műveletei, használata.
4. Az SQL nyelv részei (MSSQL specifikusan).
5. Relációs adatbázisok tervezése.
6. NoSQL adatbázisok.

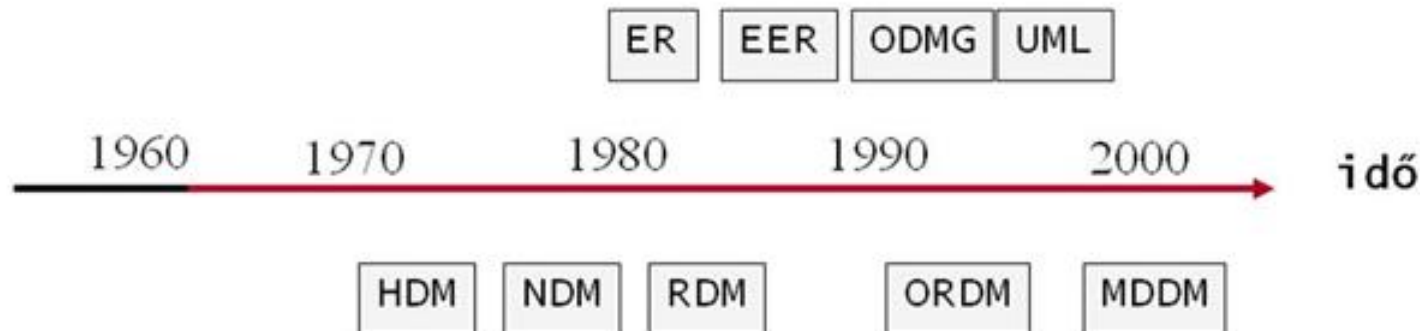
Adatmodellek

- **Adatmodell = valóság fogalmainak, kapcsolatainak, tevékenységeinek magasabb szintű ábrázolása**
- Minden ABKR egy absztrakt adatmodellel dolgozik annak érdekében, hogy az adatokat ne csak bitek sorozataként lássuk.
- Adatmodell - egy matematikai formalizmus, mely áll:
 - egy jelölés az adat leírása érdekében
 - műveletek halmaza, mely az illető adatok kezelésére használatos



Adatmodellek osztályozása

Magas szintű vagy szemantikai (SDM) vagy koncepcionális (conceptual) adatmodellek (<i>AKBR-től független</i>)	Egyed-kapcsolat (ER-Entity-Relationship) adatmodell
	Kiterjesztett egyed-kapcsolat (EER-Enhanced Entity-Relationship) adatmodell
Alacsony szintű vagy ABKR közeli adatmodellek (<i>ABKR-től függő</i>)	Hierarchikus modell
	Hálós modell
	Relációs modell
	Objektum-orientált modell



Adatmodellek osztályozása

Magas szintű vagy szemantikai (SDM) vagy koncepcionális (conceptual) adatmodellek (AKBR-től független)	Egyed-kapcsolat (ER-Entity-Relationship) adatmodell
	Kiterjesztett egyed-kapcsolat (EER-Enhanced Entity-Relationship) adatmodell
Alacsony szintű vagy ABKR közeli adatmodellek (ABKR-től függő)	Hierarchikus modell
	Hálós modell
	Relációs modell
	Objektum-orientált modell

- Szemantikai (magas szintű) adatmodell – emberközeli, lényegretörő, részletek nélküli leírása az adatoknak ↔ pontatlan
- ABKR közeli (alacsony szintű) adatmodellek – gépközel, megadja a részleteket ↔ teljes; de: túlságosan távol vannak a modellezett valóság közvetlen leírásától

→ Az elkészült szemantikai adatmodellt konvertáljuk át ABKR közeli adatmodellre.

Hierarchikus adatmodell

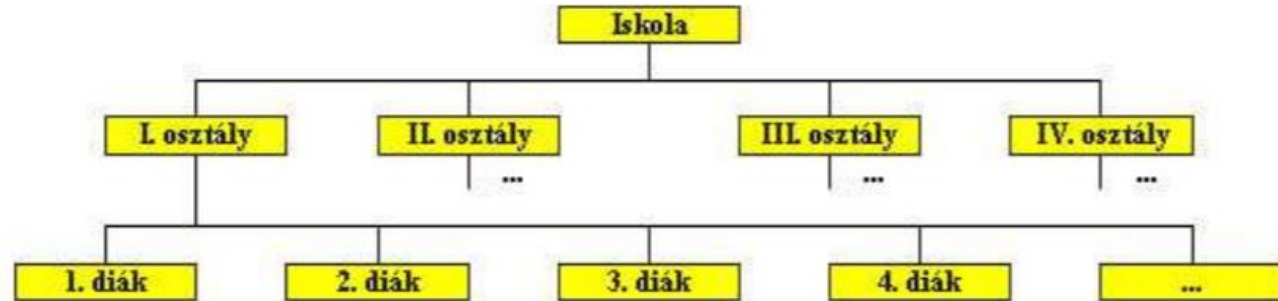
- Az 1970-es évek végéig számított elterjedtnek.
- Fa struktúrával szemléltethető.
- Az adatok között fennálló kapcsolat

szülő-gyermek

(1:N)

kapcsolatnak

felel meg.



Iskola hierarchikus felépítése a diákok szemszögéből



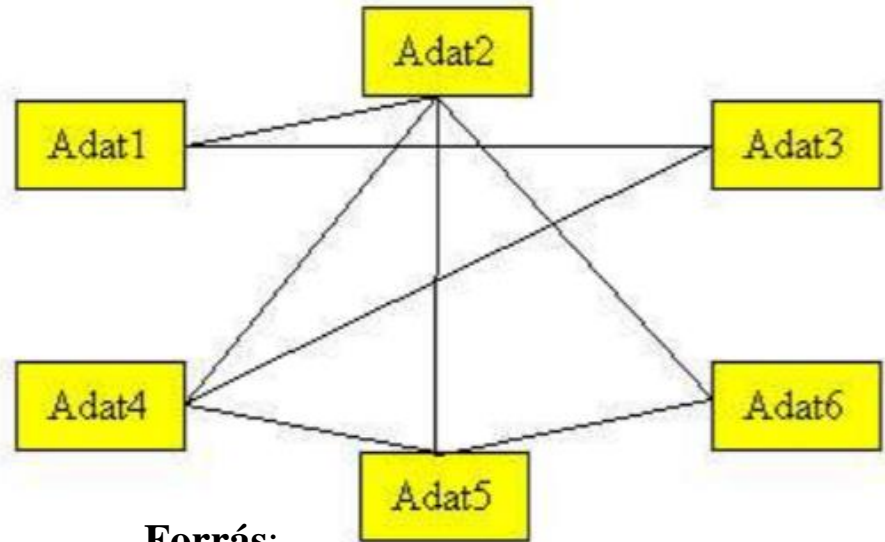
Iskola hierarchikus felépítése a tanárok szemszögéből

Forrás:

https://regi.tankonyvtar.hu/hu/tartalom/tamop425/0027_INF9/ch01s02.html

Hálós adatmodell

- Az 1970-es évek elején terjedt el.
- Az adatok közötti kapcsolatokat gráfok írják le.



Forrás:

https://regi.tankonyvtar.hu/hu/tartalom/tamop425/0027_INF9/ch01s02.html

Relációs adatmodell

- Jelenleg a **legszélesebb** körben használatos adatmodell.
- 1980-as években kezdte meg térhódítását. Halmazelméleti alapokra épülő modell.



Relációs adatmodell

Sörök

név	ország
Soproni	Magyar
Kozel	Cseh
Dreher	Német

Sörözők

név	város	tulaj
Makk 7-es	Budapest	Géza
Lórúgás	Győr	Ica

Felhasználók

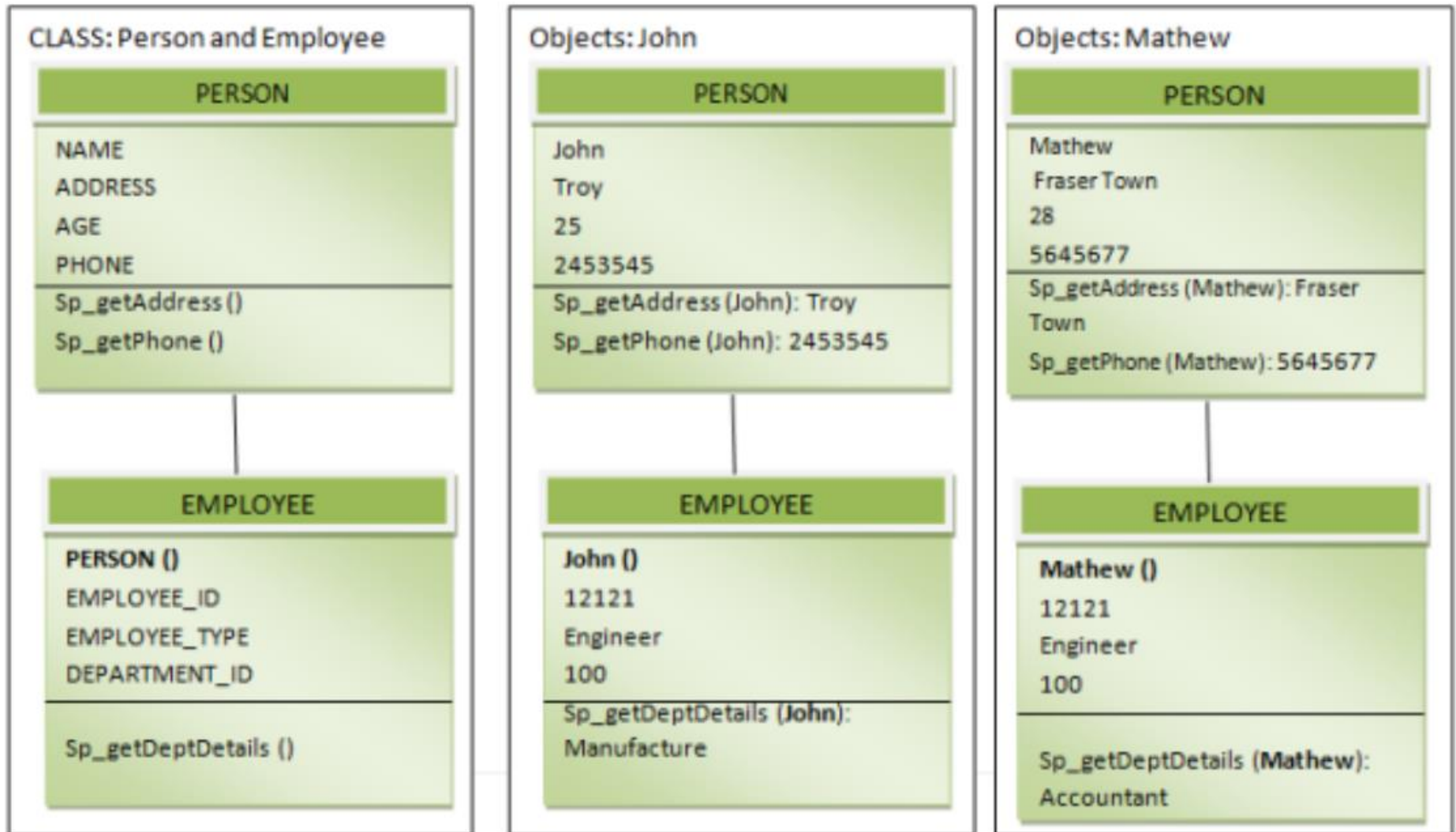
sör	bár	ár
Kozel	Makk 7-es	320
Dreher	Makk 7-es	400
Soproni	Lórúgás	280



Objektumorientált adatmodell

- Az objektumorientált programozás módszertanának egy része; az objektumorientált programozási nyelvek térhódításával terjedt el.
- **Objektumorientált tervezés** esetén a valós világot objektumok segítségével modellezzük.
- Osztály fő jellemzői: attribútum, kapcsolat, metódus.
- Az objektumorientált adatbázisok hatékonyságukban jelenleg még alulmaradnak a relációs adatbázisokkal szemben.

Objektumorientált adatmodell



Félig struktúrált adatmodell

- Relációs adatmodell korlátaira épít: a táblákba kényszerítés helyett az adatok maguk írják le a tartalmukat és a "szerkezetüket". → tetszőlegesen változó séma az idő elteltével és ugyanazon adatbázison belül is.

Unstructured data

The university has 5600 students.
John's ID is number 1, he is 18 years old and already holds a B.Sc. degree.
David's ID is number 2, he is 31 years old and holds a Ph.D. degree. Robert's ID is number 3, he is 51 years old and also holds the same degree as David, a Ph.D. degree.

Semi-structured data

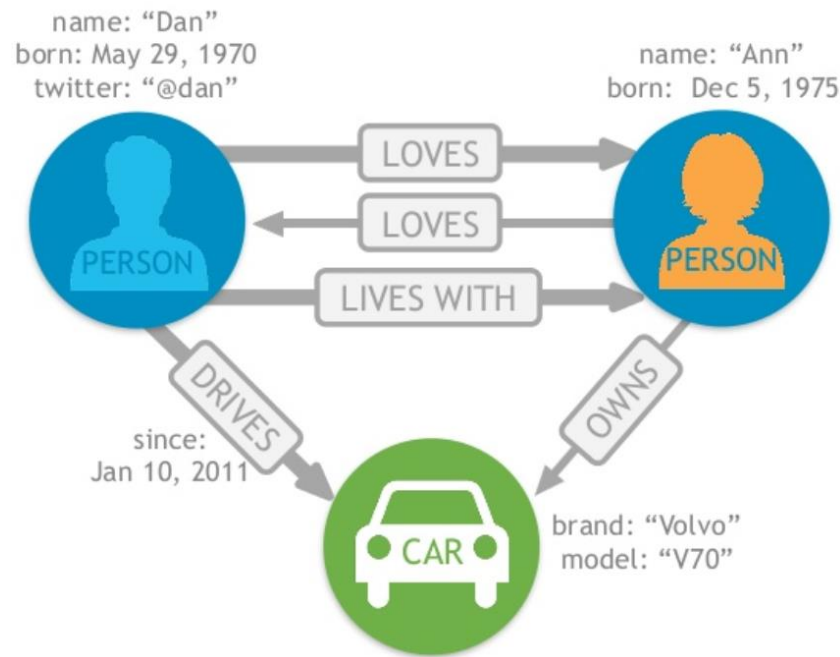
```
<University>  
  <Student ID="1">  
    <Name>John</Name>  
    <Age>18</Age>  
    <Degree>B.Sc.</Degree>  
  </Student>  
  <Student ID="2">  
    <Name>David</Name>  
    <Age>31</Age>  
    <Degree>Ph.D. </Degree>  
  </Student>  
  ....  
</University>
```

Structured data

ID	Name	Age	Degree
1	John	18	B.Sc.
2	David	31	Ph.D.
3	Robert	51	Ph.D.
4	Rick	26	M.Sc.
5	Michael	19	B.Sc.

Gráf adatmodell

- Adattárolás \leftrightarrow gráf segítségével (jellemzően: tulajdonsággráf)
- Tulajdonsággráf \leftrightarrow adatok rendszerezése csomópontokban, élekben és tulajdonságokban



Egyed-kapcsolat adatmodell

Egyed/kapcsolat adatmodell

- Megjelenés: **1976** (P. Chen)
Szabványosítás (ANSI): 1980-as évek közepe.
- **Grafikus formában** ábrázolja az **adatbázis szerkezetét**.
- **Nem teljes adatmodell** (tartalmaz néhány megszorítást, de **műveleteket nem**), de: **egyszerű grafikus jelölésrendszere** révén elterjedt és ma is széles körben használatos.
- **Magas fokú adatfüggetlenséget biztosít** \leftrightarrow *adatbázis-tervezőnek nem kell foglalkoznia az adatbázis fizikai szerkezetével.*

Egyed/kapcsolat adatmodell

- Építőelemek:
 - **Egyedhalmaz:** minden mástól elkülöníthető, azonosítható egyedek, entitások gyűjteménye
 - **Tulajdonságok:** egyedeket tulajdonságaik jellemeznék
 - **Kapcsolatok:** egyedek között kapcsolatok észlelhetők
- Ábrázolástechnika: egyed/kapcsolat diagram.
 - Két típusú jelölési rendszer/mód:
 - **Chow's feet, Chen**

Egyed/kapcsolat adatmodell építőelemei

Egyed (entitás)

- a valós világban létező, logikai vagy fizikai szempontból saját léttel rendelkező dolog (amelyről adatokat szeretnénk tárolni) \leftrightarrow konkrét dolgok absztrakciója
- Pl. hely(szín), személy (diák, alkalmazott stb.), fogalom (kurzus, pozíció, munkakör stb.), tárgy (autó, könyv stb.), érzelem (szeretet, gyűlölet), ~~hangya, tojás~~

Van-e olyan eset, amikor pl. a hangyákat külön entitásoknak tekinthetjük?

Egyed/kapcsolat adatmodell építőelemei

Egyedhalmaz

- Hasonló tulajdonságokkal rendelkező egyedek halmaza.
- Pl. járművek, vöröshajú személyek, pilóták, városok stb.
- Megnevezés: *főnevek használata*

Egyed \leftrightarrow objektum, egyedhalmaz \leftrightarrow osztály \leftarrow **objektum-orientált adatmodell**

- Példa a jelölésre:

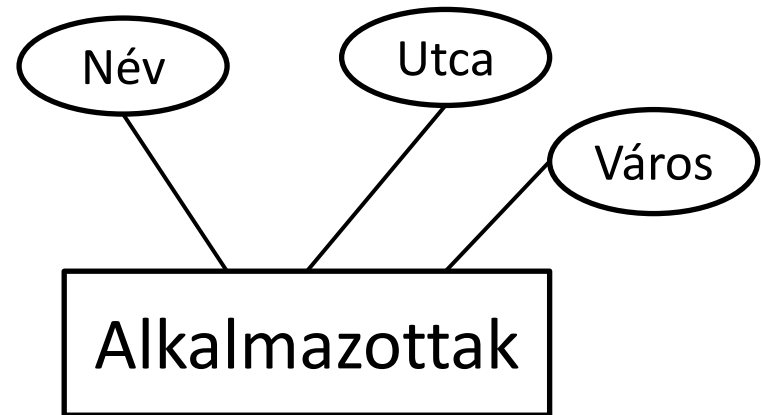
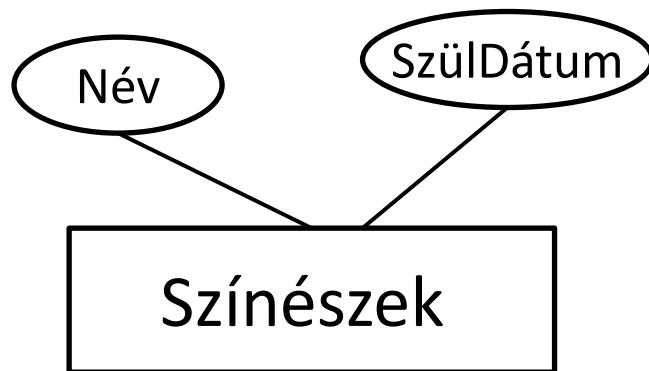
Színészek

Alkalmazottak

Egyed/kapcsolat adatmodell építőelemei

Tulajdonságok (attribútumok)

- Azon jellemző(k), amelye(ke)n keresztül az entitások megkülönböztethetők egymástól.
- Értékei egy egyed tulajdonságát írják le.
- Pl. személy neve, címe; jármű márkája, színe stb.
- Példa a jelölésre:



Egyed/kapcsolat adatmodell építőelemei

Általános jelölés

- $E(A_1, \dots, A_n)$ egy **egyedhalmaz sémája**, ahol:
 - E – egyedhalmaz neve,
 - A_1, \dots, A_n – attribútumok.
 - $\text{dom}(A_i)$ – a lehetséges értékek halmaza
 - Legegyszerűbb esetben: csak atomi értékek
 - Bonyolultabb esetek: pl. összetett- vagy többértékű tulajdonság
 - **Példa:** CsokoládéGyárak (Név, Cím, Helység)
Filmek (FilmCím, MegjÉv, Stúdió)
Tanárok (Név, Tanszék)

Egyed/kapcsolat adatmodell építőelemei

Általános jelölés

- $E(A_1, \dots, A_n)$ sémájú egyedhalmaz előfordulása:
 - A konkrét egyedekből áll.
 - $E = \{e_1, \dots, e_m\}$ egyedek (entitások) halmaza, ahol:
 $e_{i_k} \in \text{Dom}(A_k), i=1, \dots, m, k=1, \dots, n.$
 - *Semelyik két egyed sem egyezik meg minden attribútumán
 \Rightarrow kulcsok*

Példa:

Alkalmazottak egyedhalmaz előfordulása

AlkID	Név	Utca	Város
321-12-3123	Jones	Main	Harrison
019-28-3746	Smith	North	Rye
677-89-9011	Hayes	Main	Harrison
555-55-5555	Jackson	Dupont	Woodside
244-66-8800	Curry	North	Rye
963-96-3963	Williams	Nassau	Princeton
335-57-7991	Adams	Spring	Pittsfield

Egyed/kapcsolat adatmodell építőelemei

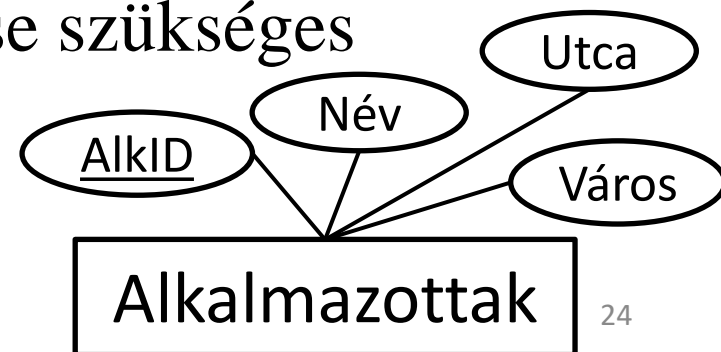
Kulcs

- egy vagy több attribútum, mely egyértelműen meghatároz egy egyedet az egyedhalmazban \leftrightarrow
semelyik két egyed sem egyezik meg minden attribútumán.
- Kikötés: minden egyed megkülönböztethető \rightarrow *minden egyedhalmaznak van kulcsa*
- *Példa?*

Egyed/kapcsolat adatmodell építőelemei

Kulcs

- egy vagy több attribútum, mely egyértelműen meghatároz egy egyedet az egyedhalmazban \leftrightarrow semelyik két egyed sem egyezik meg minden attribútumán
- Kikötés: minden egyed megkülönböztethető \rightarrow *minden egyedhalmaznak van kulcsa*
- Pl. autó – rendszám, személy – személyazonosító száma stb.
- Gyakran **plusz attribútum** bevezetése szükséges (pl. FilmID/SzineszID, TeremKod)
- Jelölés: aláhúzással

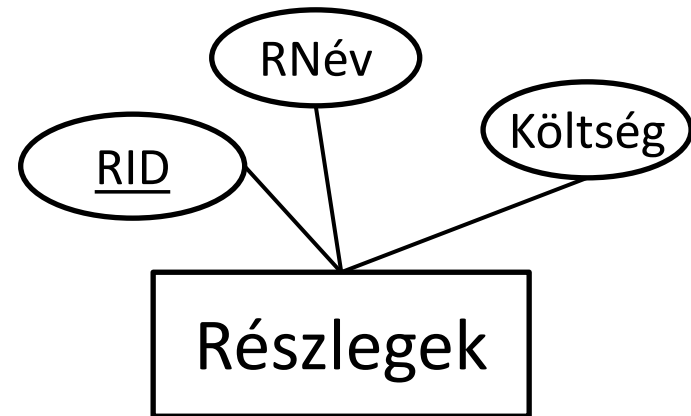
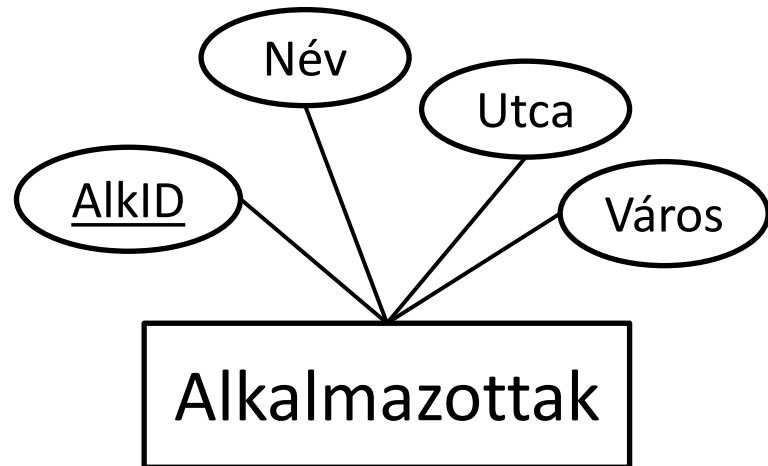


Egyed/kapcsolat adatmodell építőelemei

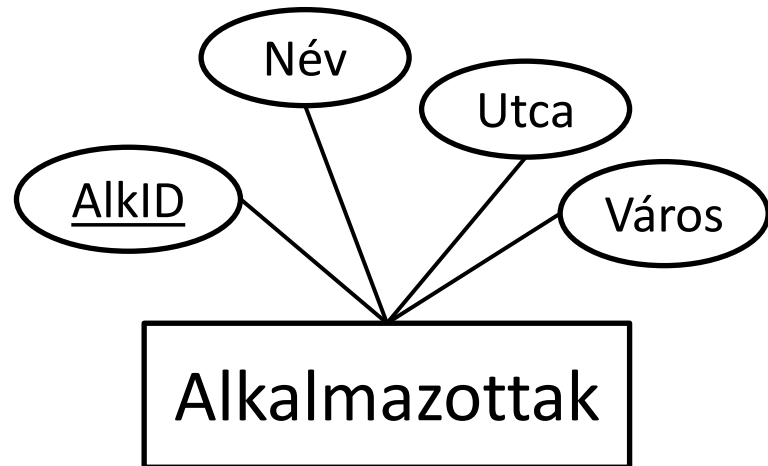
Kapcsolatok

- egyedhalmazok előfordulásai közötti relációk
- Megnevezésben: *igék, melléknevek használata*
- Pl.
 - Tanít – kapcsolat Tanárok és Diákok között
 - Vásárol – kapcsolat Vevők és Eladók között
 - Kezel – kapcsolat Betegek és Orvosok között
- Kapcsolat foka (jel. k):
 - $k = 2 \rightarrow$ bináris kapcsolat
 - $k > 2 \rightarrow$ többágú kapcsolat

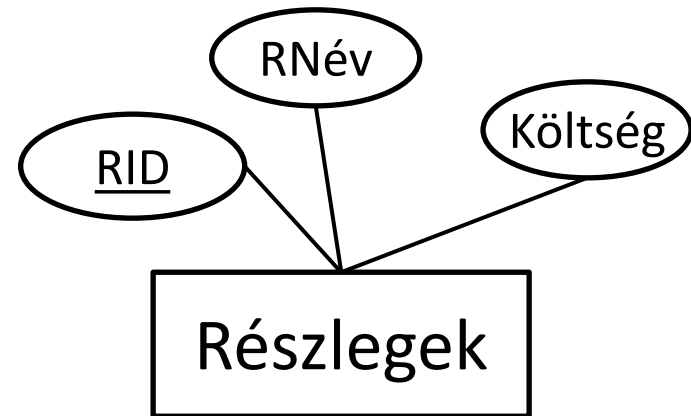
Példa



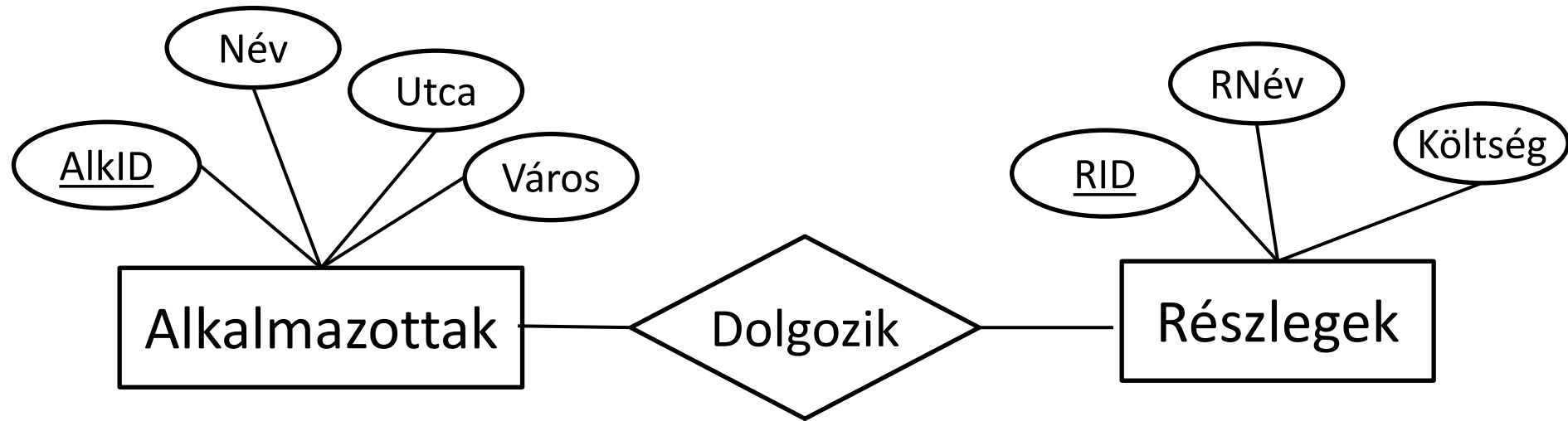
Példa



Kapcsolat?



Példa

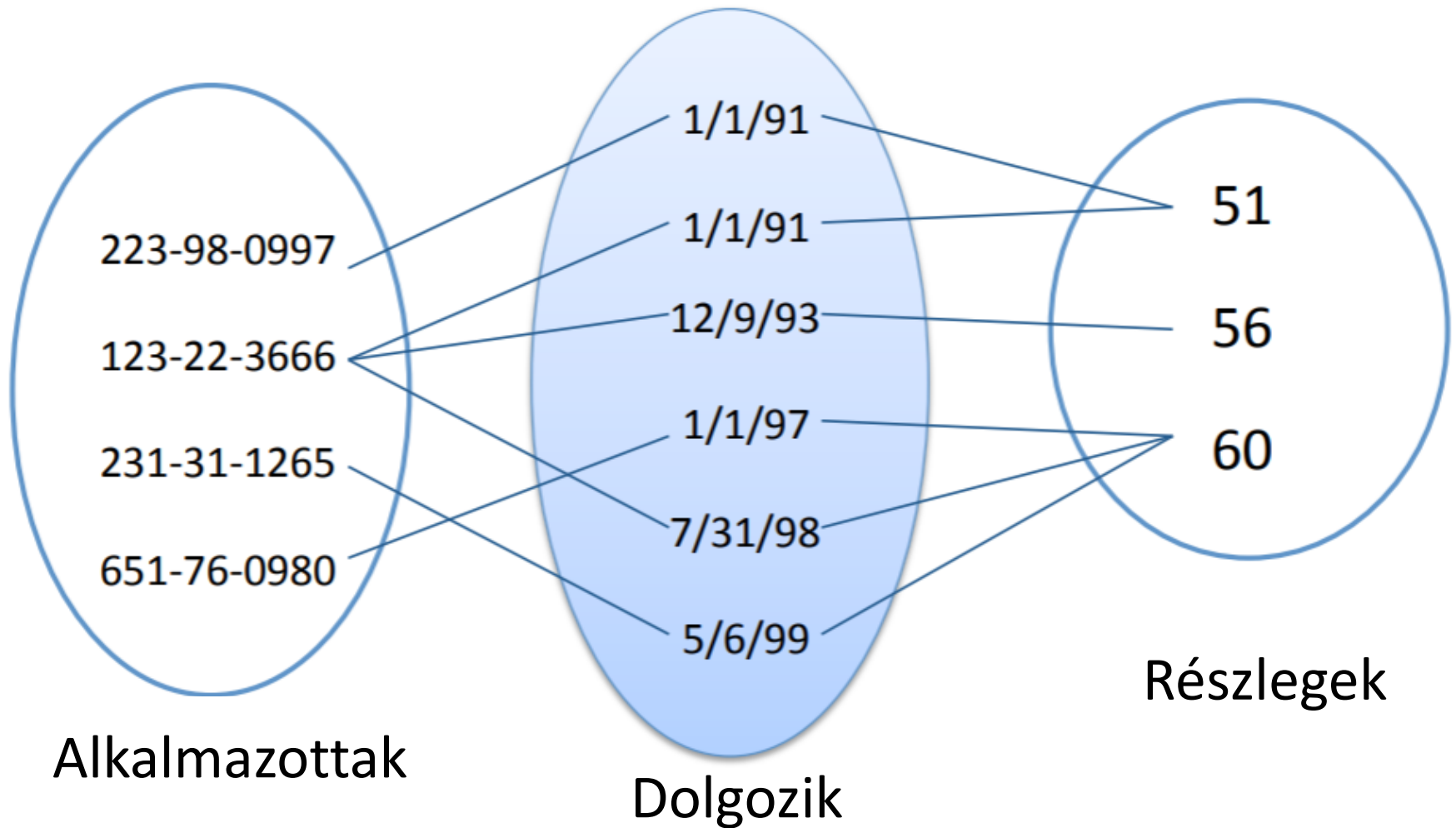


Egyed/kapcsolat adatmodell építőelemei

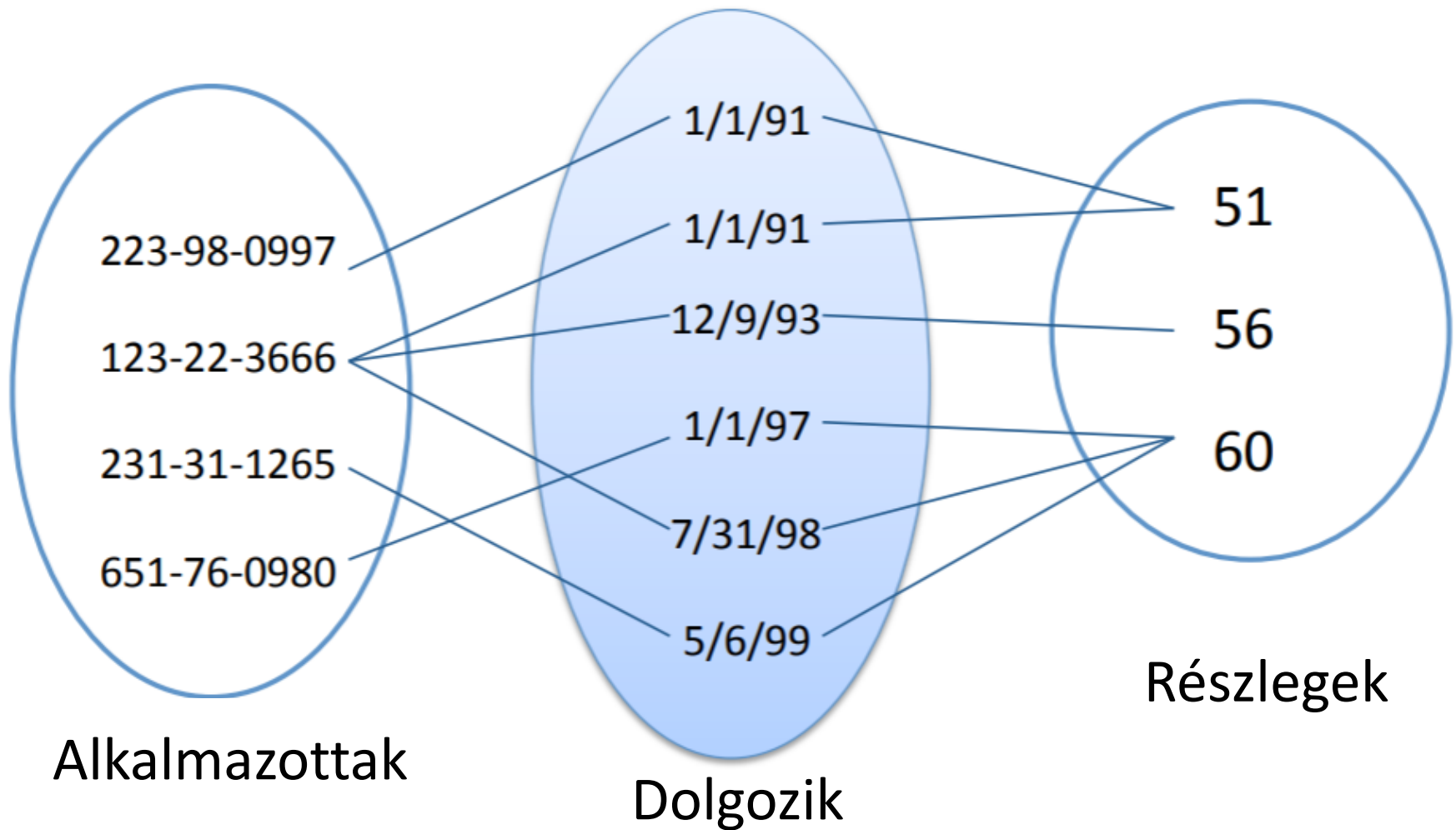
Általános jelölés

- $K(E_1, \dots, E_k, A_1, \dots, A_m)$ egy **kapcsolat sémája**, ahol:
 - K a kapcsolat neve,
 - E_1, \dots, E_k egyedhalmazok sémái,
 - A_1, \dots, A_m a kapcsolathoz tartozó attribútumok.
 - k – **kapcsolat foka**
 - $k=2 \rightarrow$ bináris kapcsolat, $k>2 \rightarrow$ többágú kapcsolat
- $K(E_1, \dots, E_k)$ sémájú kapcsolat egy **előfordulása**:
 (e_1, \dots, e_k) , ahol $e_i \in E_i, i=1, \dots, k$.

Példa (folyt.)

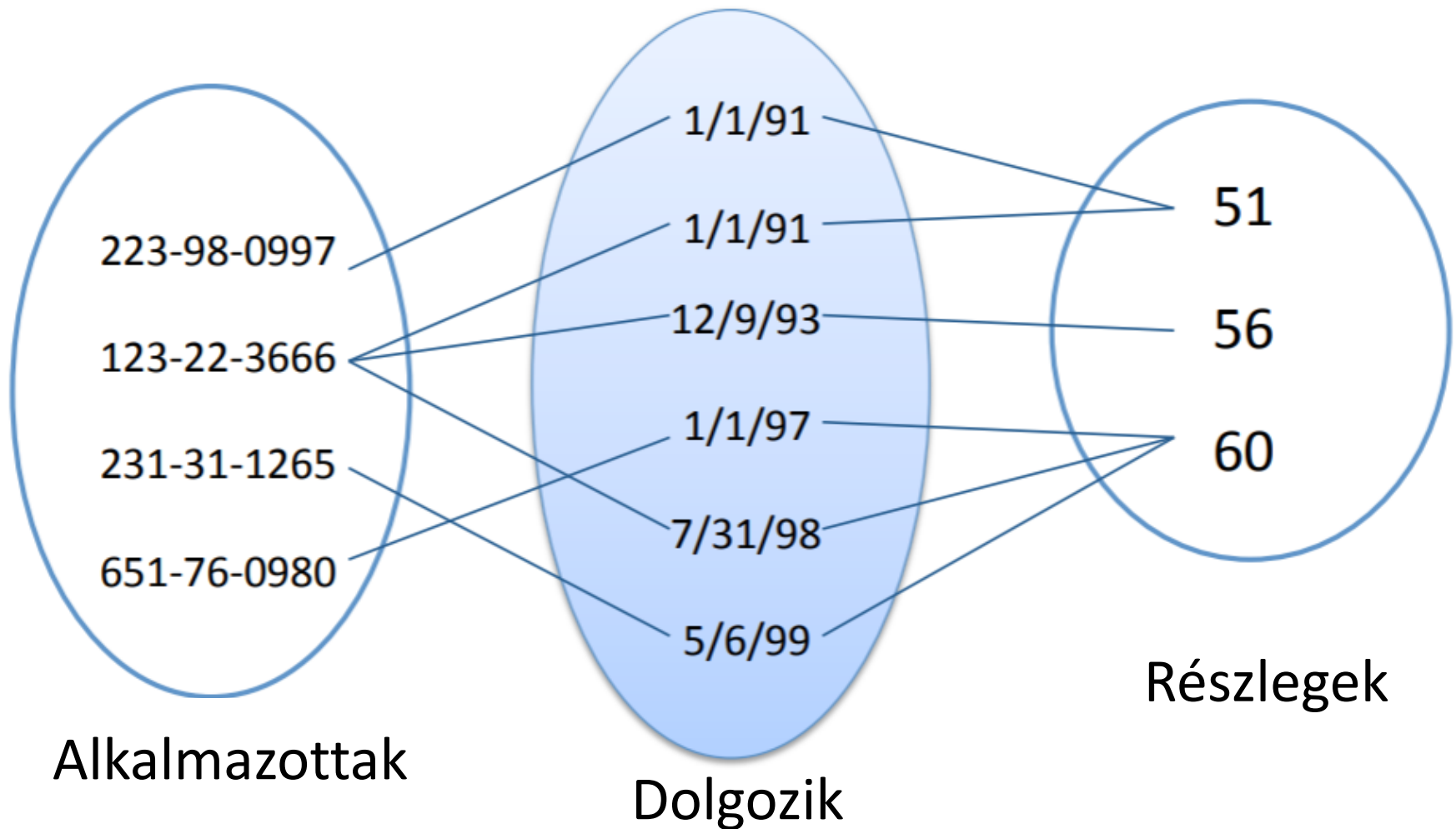


Példa (folyt.)



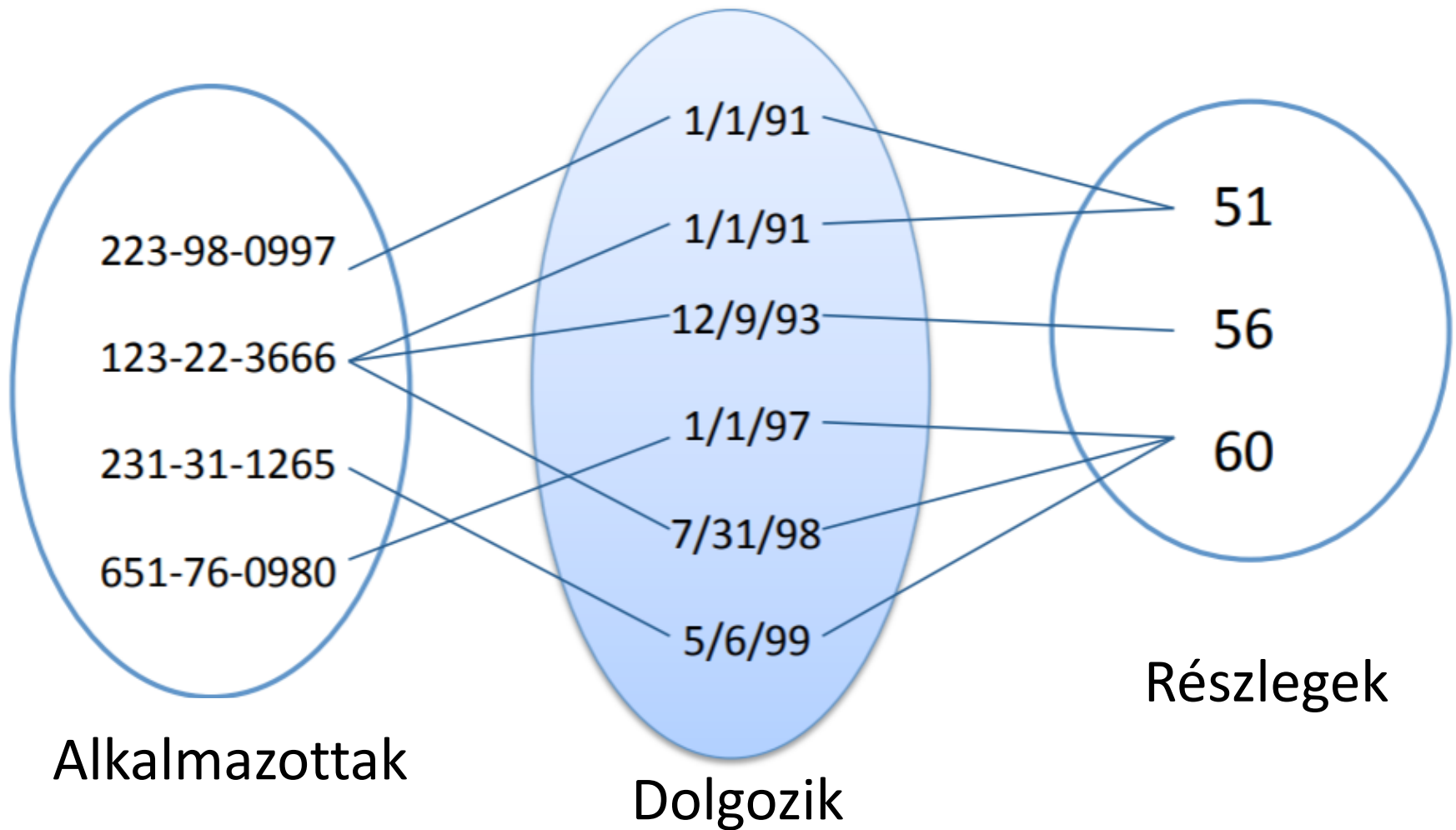
Kapcsolat előfordulás:
(223-98-0997, 51, 1/1/91)

Példa (folyt.)



*Milyen típusú a Dolgozik kapcsolat?
Lehetett volna-e másként ábrázolni?*

Példa (folyt.)



- *Kapcsolat típusának meghatározása - a kapcsolat előfordulásaira tett megszorítások által*

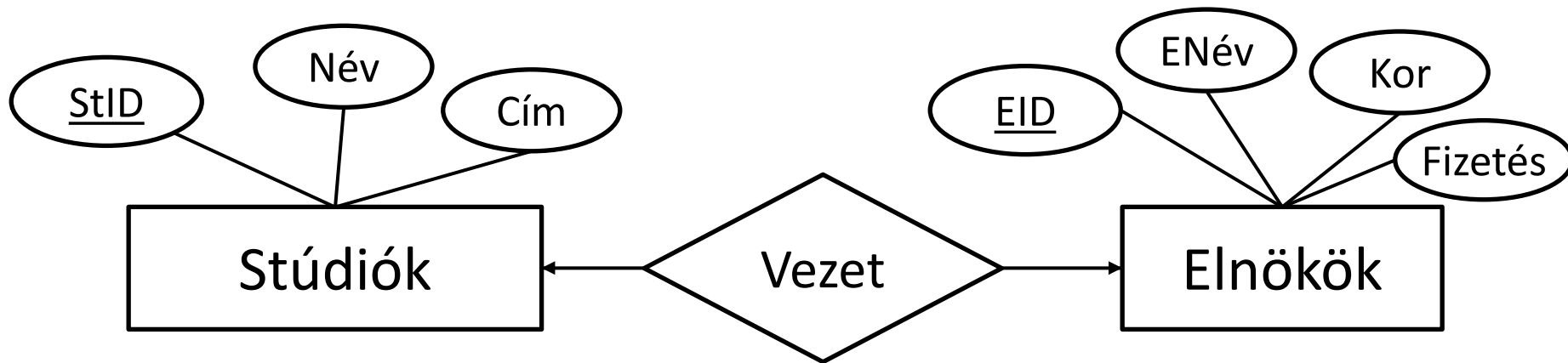
Kapcsolatokra vonatkozó megszorítások

- **Számosság:** A (bináris) kapcsolatok *számossága* meghatározza azon kapcsolat-előfordulások maximális számát, amelyekben egy egyed részt vehet.
 - A kapcsolat előfordulásaira tett megszorítások határozzák meg a *kapcsolat típusát*.
- A bináris kapcsolattípusok lehetséges számosságai:
 - Egy az egyhez (1:1)
 - Egy a többhöz (1:N); Több az egyhez (N:1)
 - Több a többhöz (M:N)
- Számosságra vonatkozó megszorítás jelölése:
 - Irányított vonal (\rightarrow) – „egy”-et jelöl; irányítatlan vonal (—) – „több”-et jelöl (a kapcsolathalmaz és az egyedhalmaz között)

Kapcsolatok típusai

- Legyen $K(E_1, E_2)$ – bináris kapcsolat
- Egy az egyhez kapcsolat (1:1)
 - Egy E_1 -beli egyedhez legfeljebb egy E_2 -beli egyed tartozhat és fordítva.
 - Pl. Osztályok-Osztályfőnökök

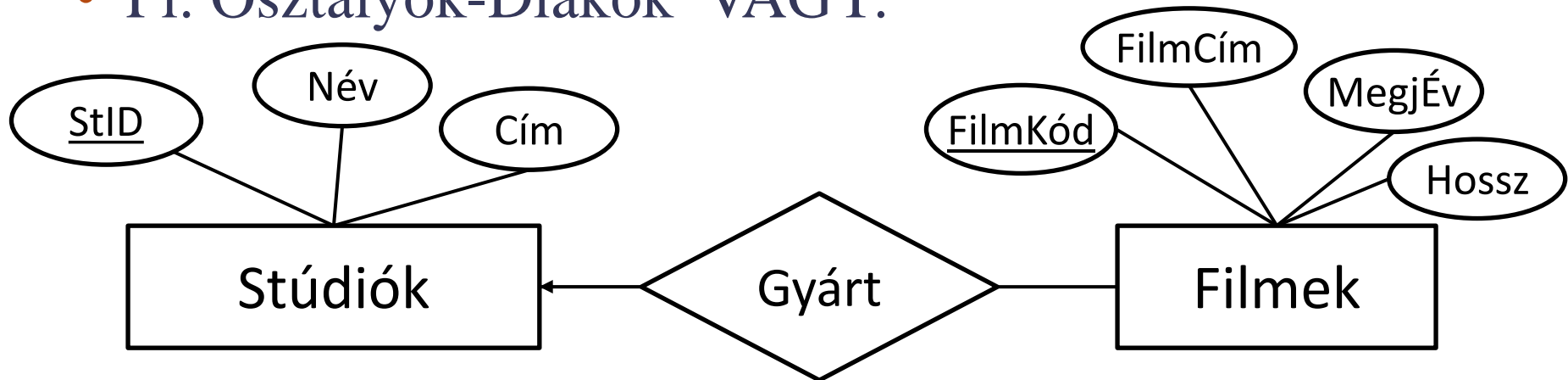
VAGY:



Kapcsolatok típusai

■ Egy a többhöz kapcsolat (1:n)

- Egy E_1 -beli egyed kapcsolatba hozható 0 vagy több E_2 -beli egyeddel, egy E_2 -beli egyed azonban legfeljebb egy E_1 -beli egyedhez tartozhat.
- Pl. Osztályok-Diákok VAGY:



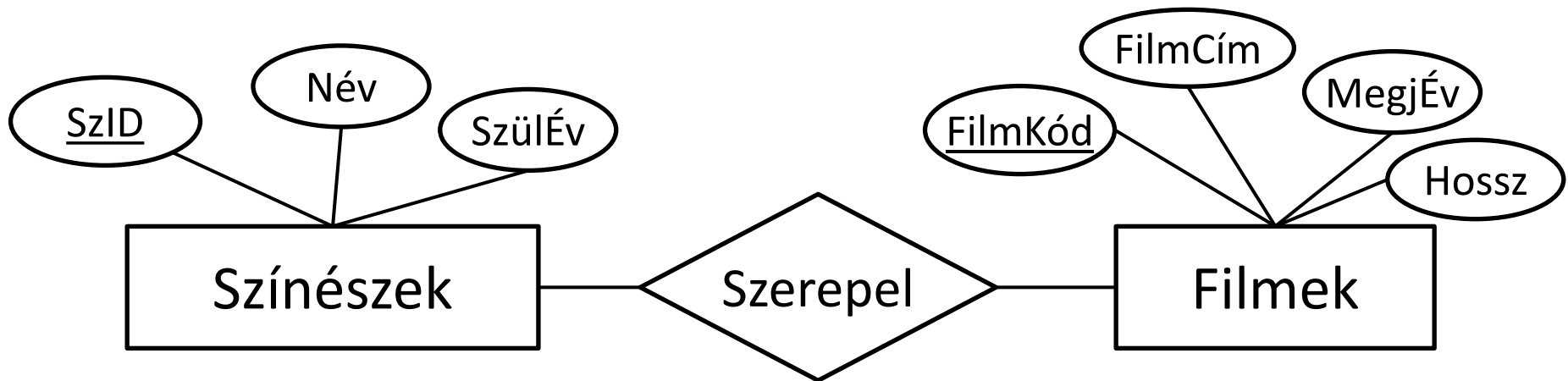
Megj.: nem kötelezően szerepel minden egyed a kapcsolatban,
pl. házaspár(férfi,nő) *[lásd később]*

Kapcsolatok típusai

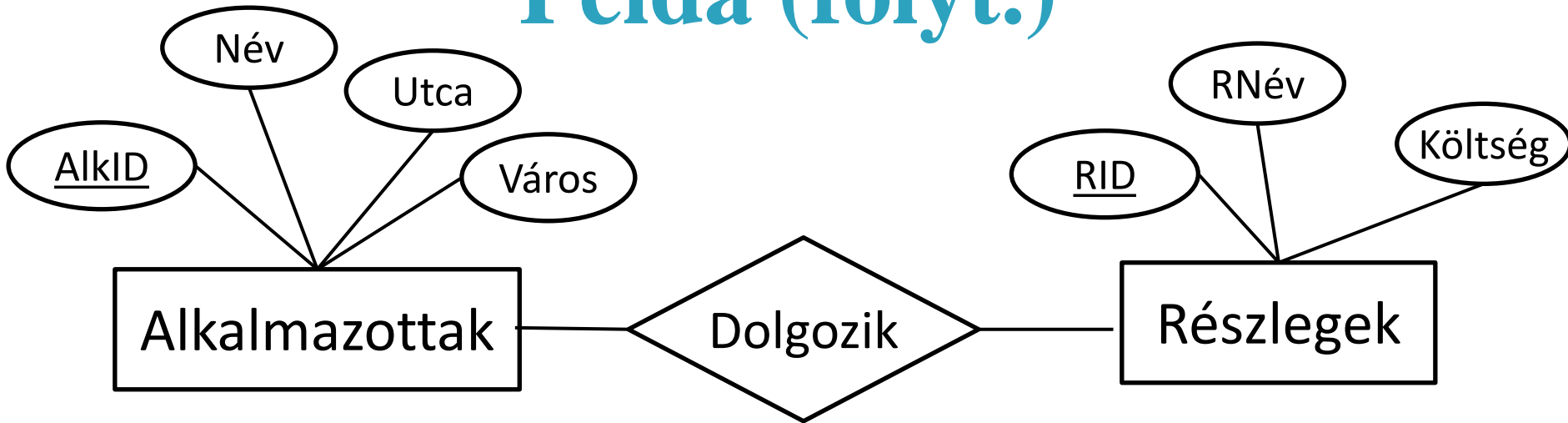
■ Több a többhöz kapcsolat (n:m)

- minden E_1 -beli egyedhez több E_2 -beli egyed tartozhat, és fordítva
- Pl. Tanárok-Tantárgyak

VAGY:

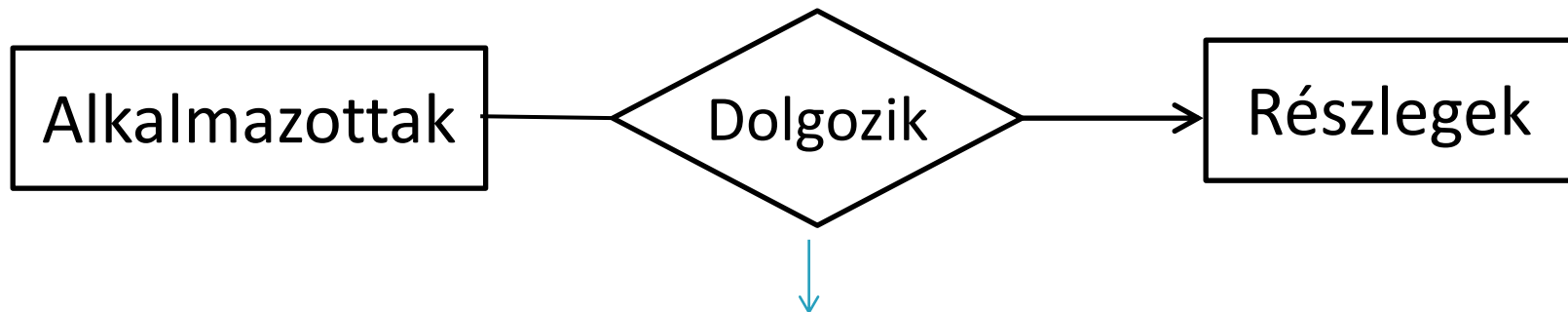


Példa (folyt.)



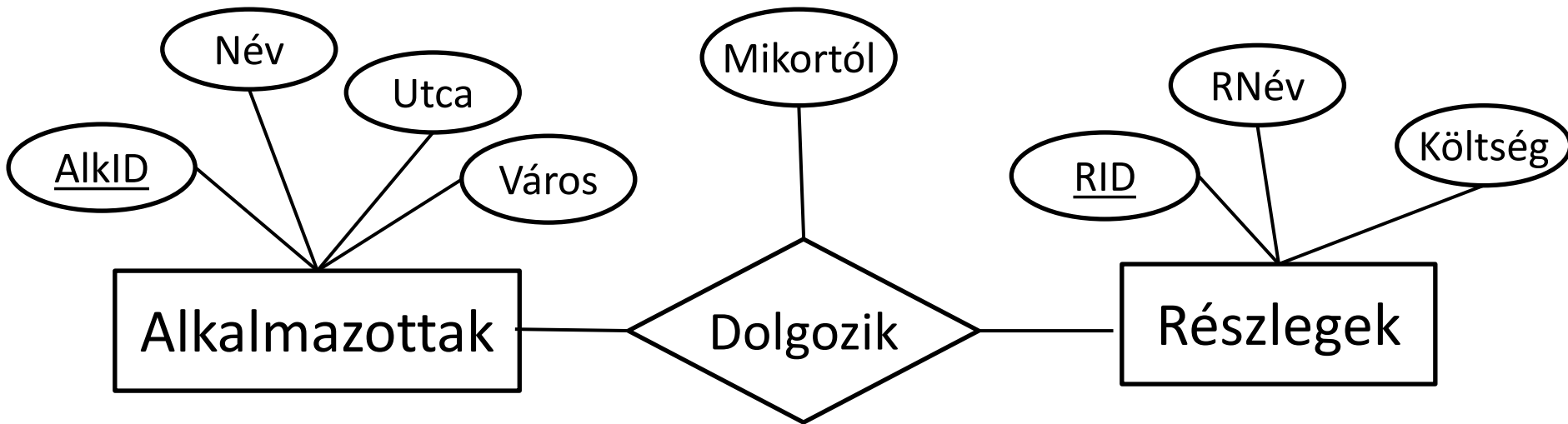
bináris, **több a többhöz (n:m) típusú kapcsolat**

Más lehetőség:



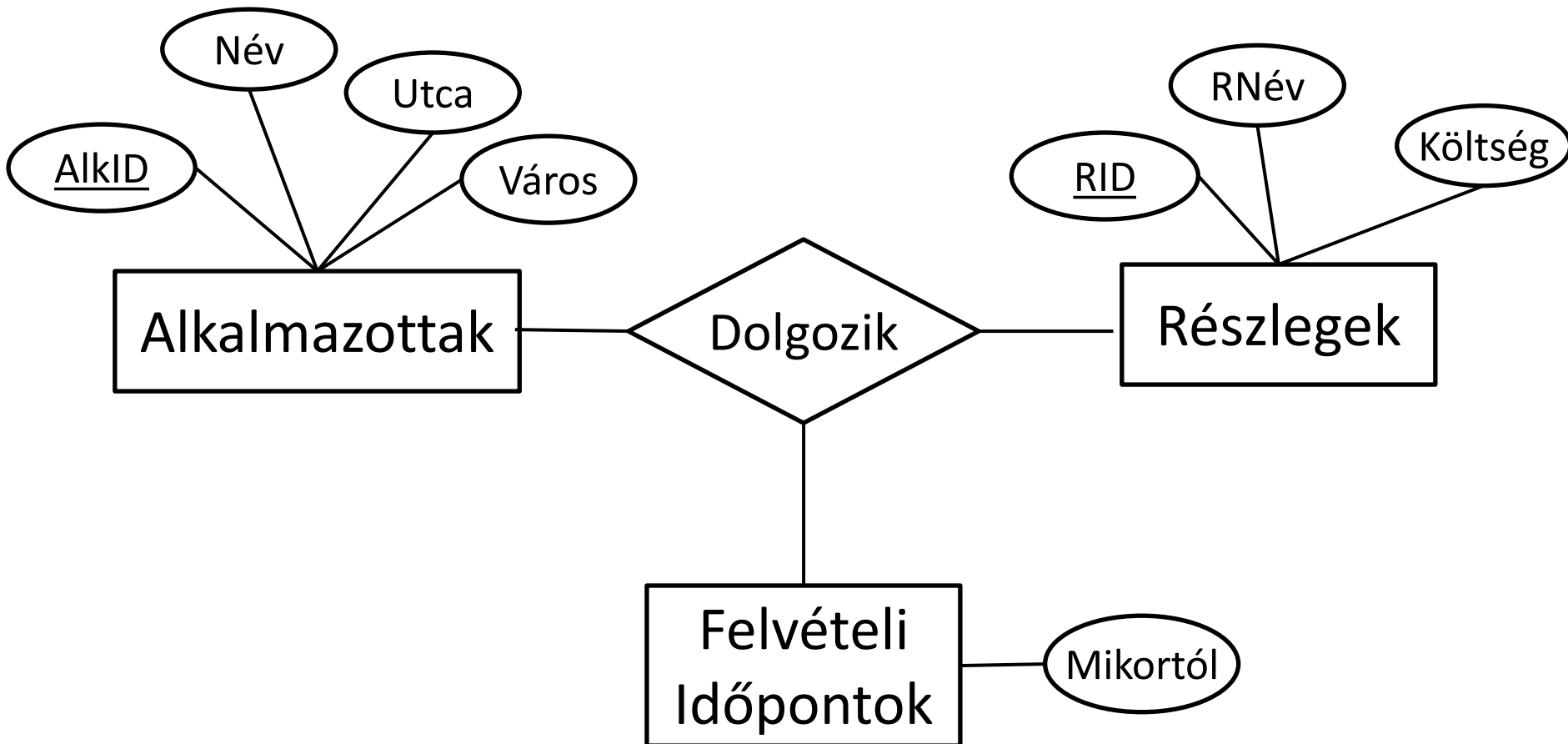
bináris, **több az egyhez (n:1) típusú kapcsolat**

Kapcsolat attribútuma



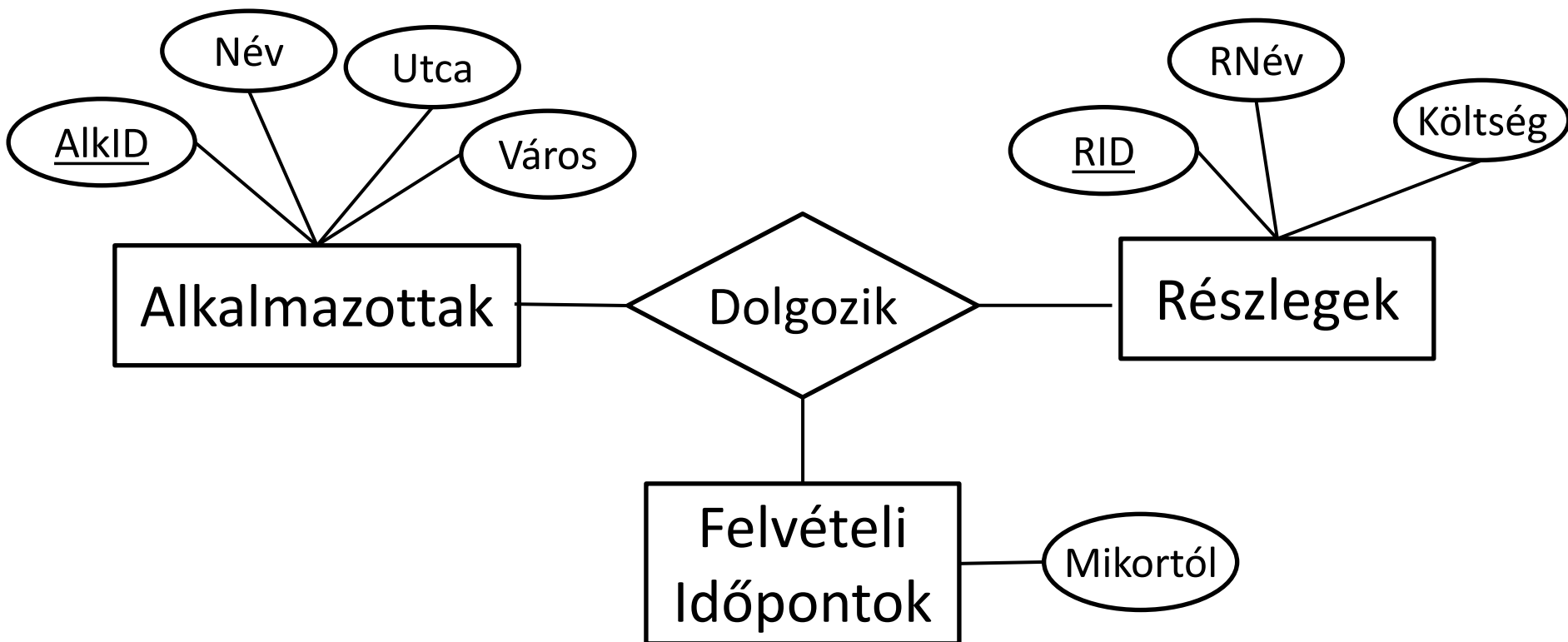
- *Mikortól: az alkalmazott és részleg együttesen határozzák meg, de egyik sem külön.*
- **Csak több a többhöz (n:m) típusú kapcsolat esetén megengedett.**

Attribútum vs. egyedhalmaz



Előnyök, hátrányok?

Attribútum vs. egyedhalmaz



Előny: Olyan értékeket (pl. időpontokat, árakat) is lehetőségünk van tárolni, amelyek még nem szerepelnek a kapcsolatban, de csak ezek a lehetséges értékek (pl. árértékek). + statisztikák készítésekor

Hátrány: megvalósítás esetén

(plusz egyedhalmaz \Leftrightarrow plusz tábla, lassabb adatlekérés).⁴¹

„Kapcsolat-különlegességek”

Unáris/rekurzív kapcsolat

- A társult/kapcsolatban levő/megjelenő egyedhalmazok száma (kapcsolat foka) = 1.
- Egyedhalmaz önmagával való kapcsolata = **rekurzív kapcsolat**
- **Példa:**

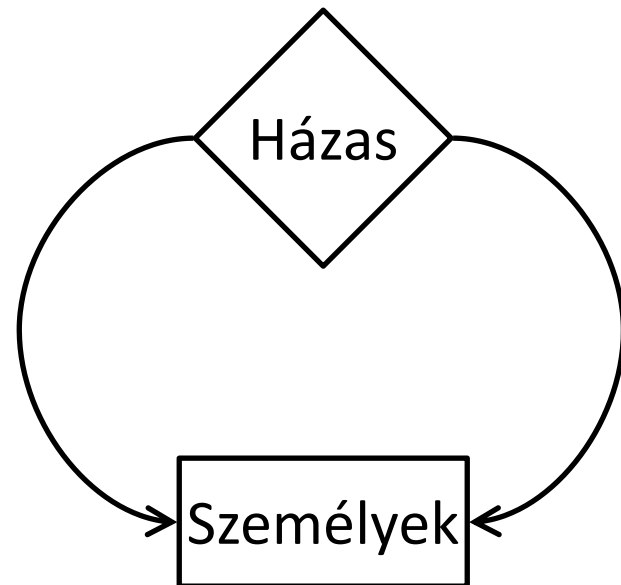
„Kapcsolat-különlegességek”

Unáris/rekurzív kapcsolat

- A társult/kapcsolatban levő/megjelenő egyedhalmazok száma (kapcsolat foka) = 1.
- Egyedhalmaz önmagával való kapcsolata = **rekurzív kapcsolat**
- **Példa:** házasság ábrázolása

Kapcsolathalmaz:

Név	Név
Jani	Fruzszi
Kati	Andris
...	...



„Kapcsolat-különlegességek”

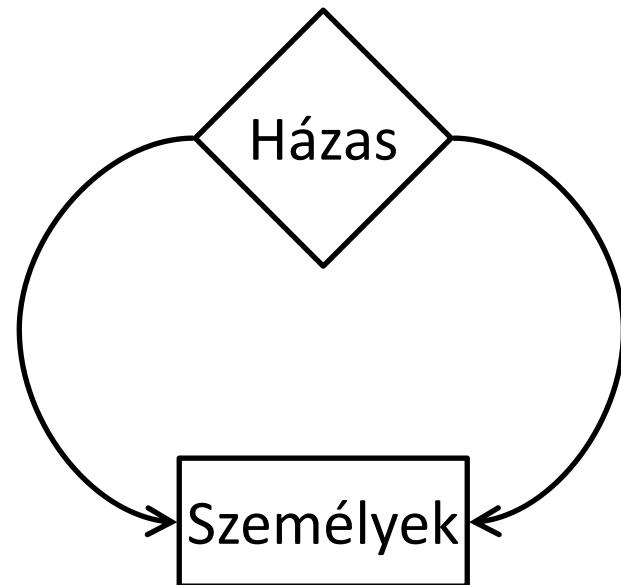
Unáris/rekurzív kapcsolat

- A társult/kapcsolatban levő/megjelenő egyedhalmazok száma (kapcsolat foka) = 1.
- Egyedhalmaz önmagával való kapcsolata = **rekurzív kapcsolat**
- **Példa:** házasság ábrázolása

Kapcsolathalmaz:

Név	Név
Jani	Fruzszi
Kati	Andris
...	...

Probléma(k)?



„Kapcsolat-különlegességek”

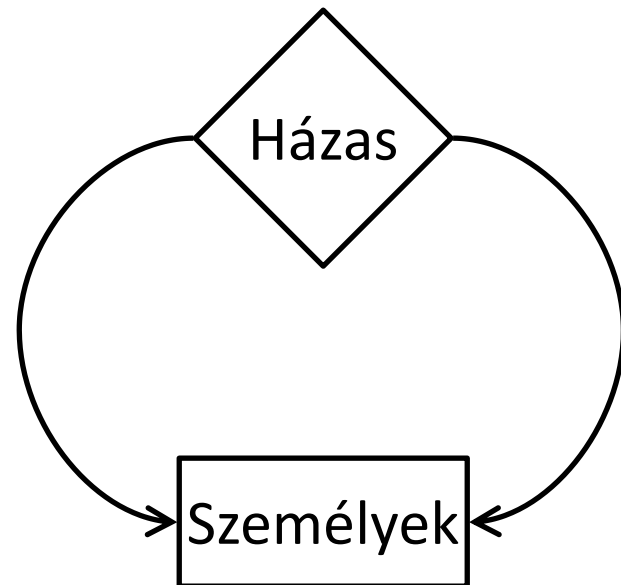
Unáris/rekurzív kapcsolat

- A társult/kapcsolatban levő/megjelenő egyedhalmazok száma (kapcsolat foka) = 1.
- Egyedhalmaz önmagával való kapcsolata = **rekurzív kapcsolat**
- **Példa:** házasság ábrázolása
- **Probléma:** *(Név, Név) séma*

nem megengedett

Kapcsolat
halmaz:

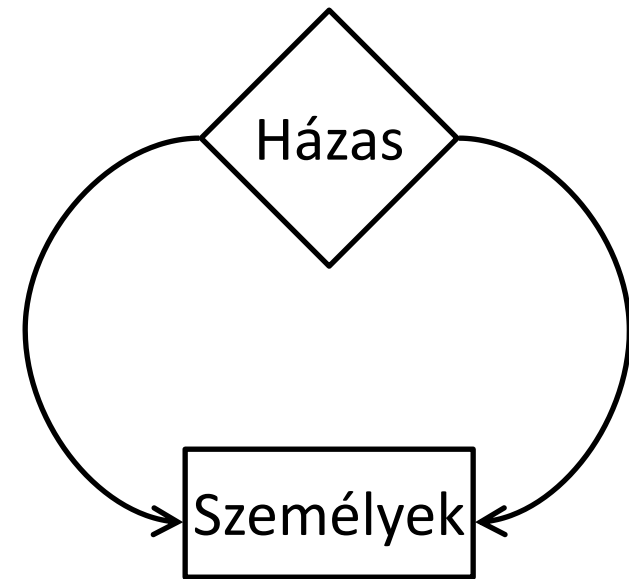
Név	Név
Jani	Fruzszi
Kati	Andris
...	...



„Kapcsolat-különlegességek”

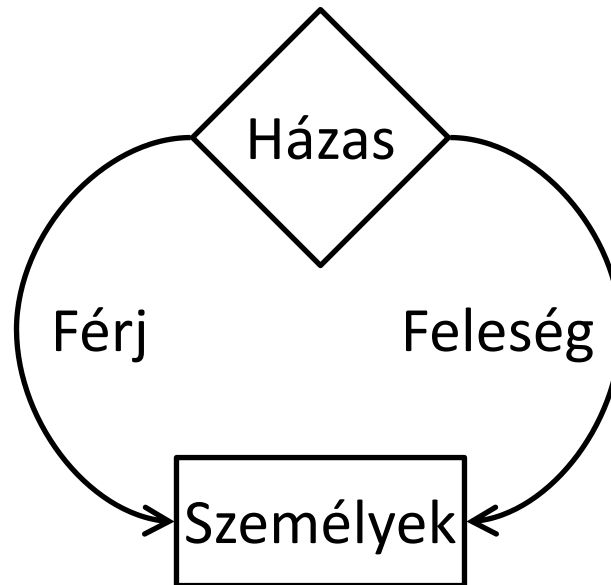
Unáris/rekurzív kapcsolat

- A társult/kapcsolatban levő/megjelenő egyedhalmazok száma (kapcsolat foka) = 1.
- Egyedhalmaz önmagával való kapcsolata = **rekurzív kapcsolat**
- **Probléma:** *nem derül ki az egyedhalmaznak a kapcsolatban játszott „szerepe”.*
 - **Példa:** házasság ábrázolása
Itt: Melyik él szimbolizálja a feleséget és melyik a férjet?
- **Másik probléma:** *két vagy több azonos attribútumnév nem megengedett ugyanazon relációsémában (lsd. (Név, Név)) (lsd. később)*



Példa unáris kapcsolatra

- **Megoldás:** élek címkézése → kapcsolat adott **szerepe**



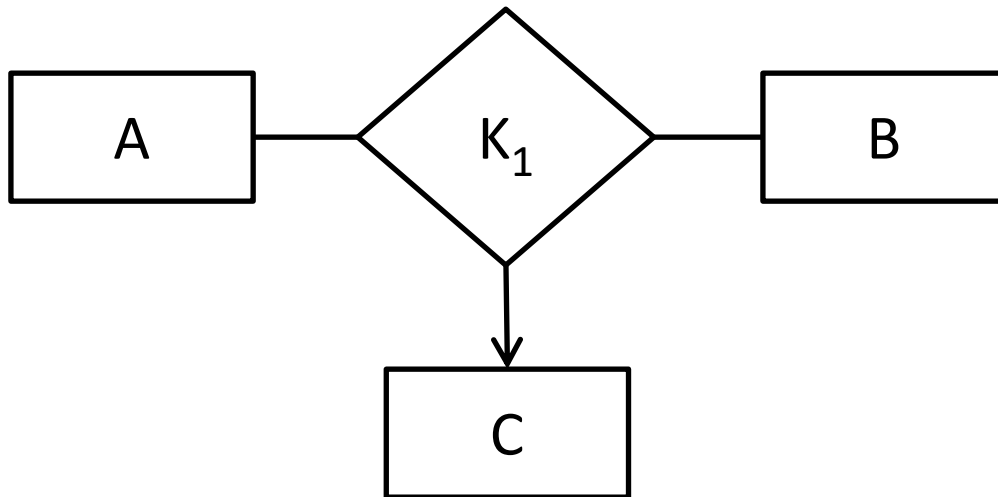
Kapcsolathalmaz:

Férj	Feleség
Jani	Fruzszi
Andris	Kati
...	...

- *Más lehetőség házasság ábrázolására?*
- *Mi határozza meg a kapcsolat ábrázolásának módját?*

„Kapcsolat-különlegességek”

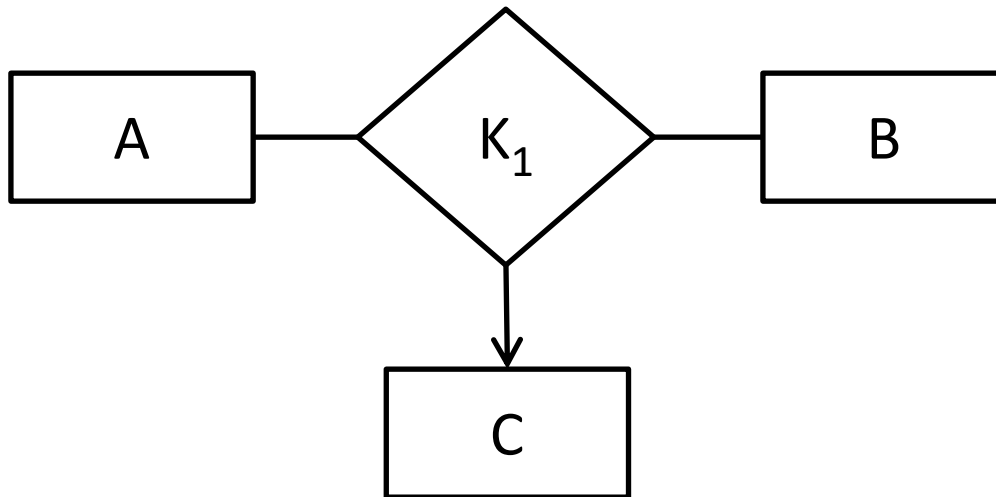
Többágú kapcsolatok



„Kapcsolat-különlegességek”

Többágú kapcsolatok

- K_1 - háromágú kapcsolat A, B és C egyedhalmazok között

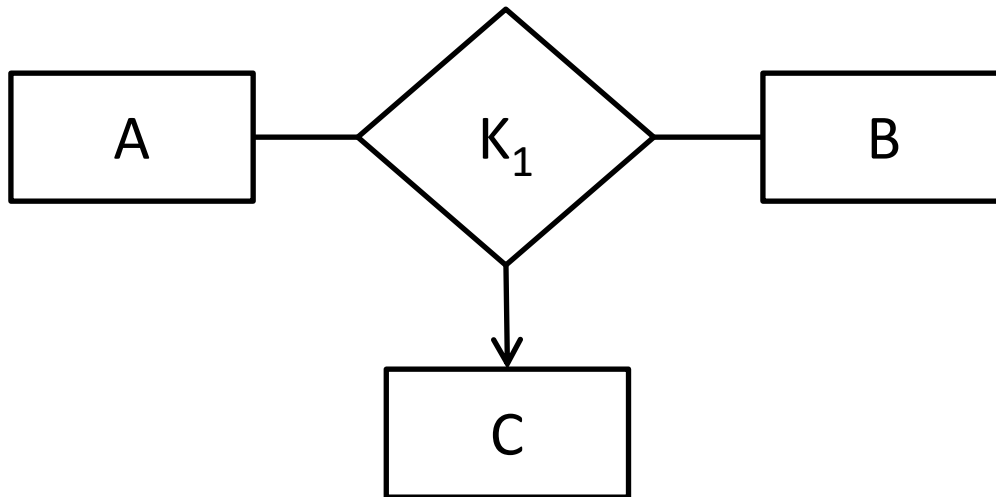


- Nyíl szerepe többágú kapcsolatoknál megváltozik

„Kapcsolat-különlegességek”

Többágú kapcsolatok

- K_1 - háromágú kapcsolat A, B és C egyedhalmazok között

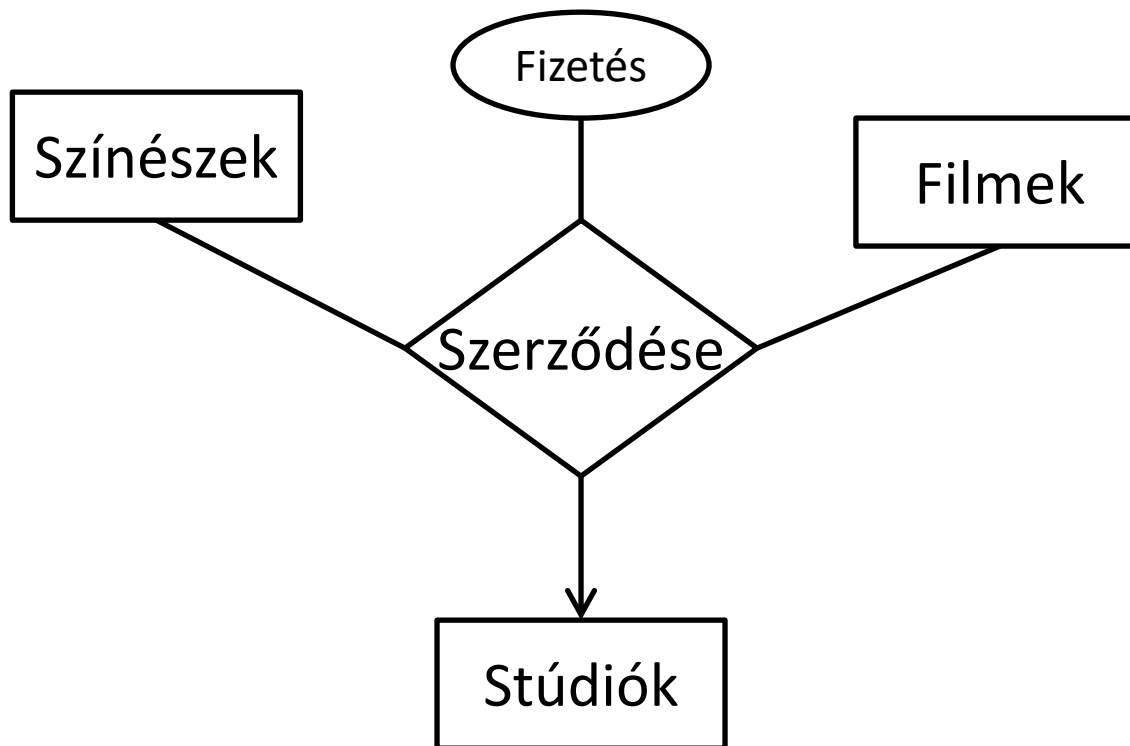


- Nyíl szerepe többágú kapcsolatoknál megváltozik: egy konkrét A-beli és egy konkrét B-beli egyed előforduláshoz csak egy C-beli előfordulás tartozik

„Kapcsolat-különlegességek”

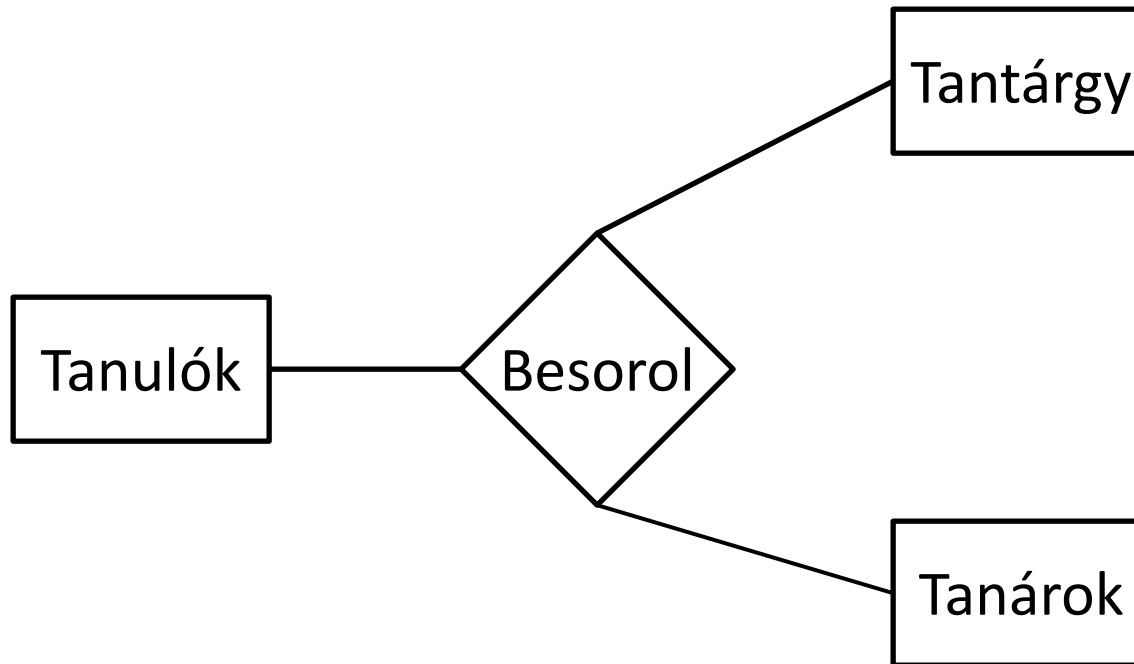
Többágú kapcsolatok

- Példák háromágú kapcsolatra:



Szerződése: 1:n:m típusú

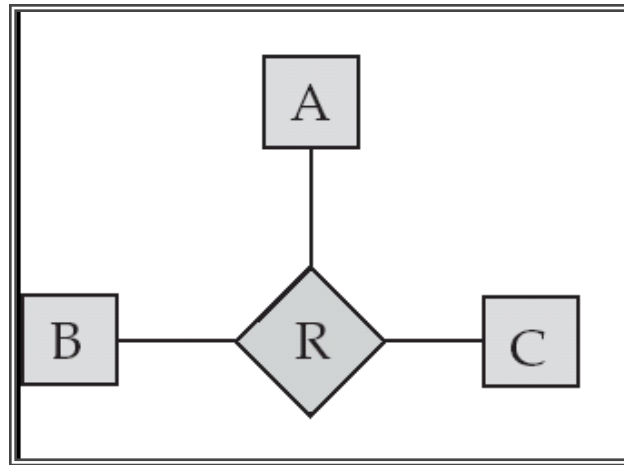
További példa többágú kapcsolatra



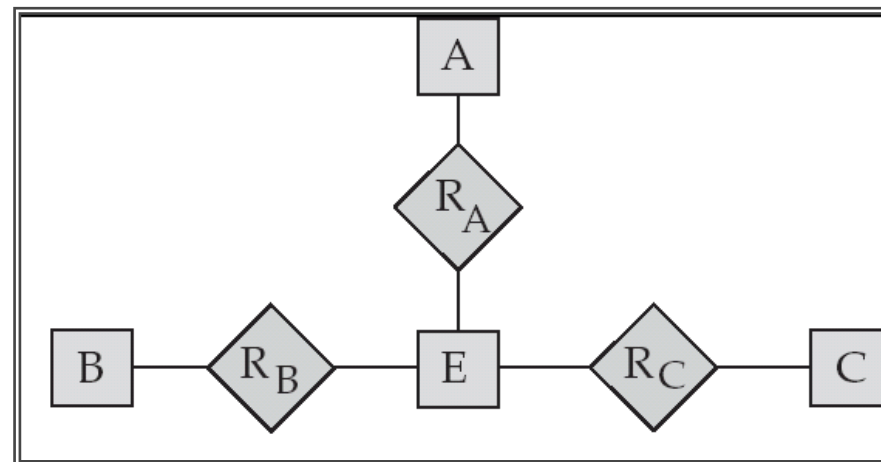
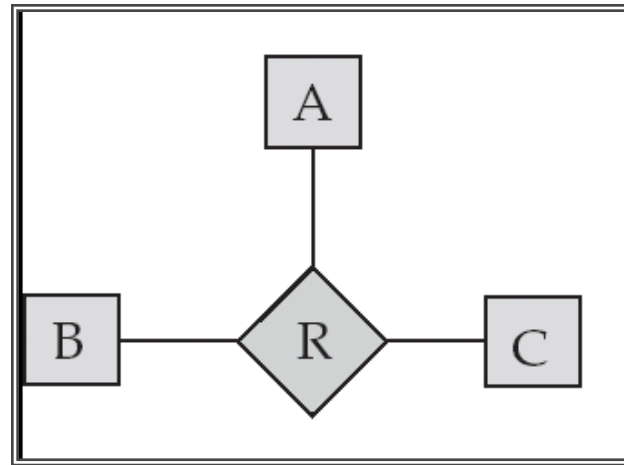
Besorol: n:m:p típusú

Más példa n-ágú kapcsolatra?

Többágú kapcsolat átírása

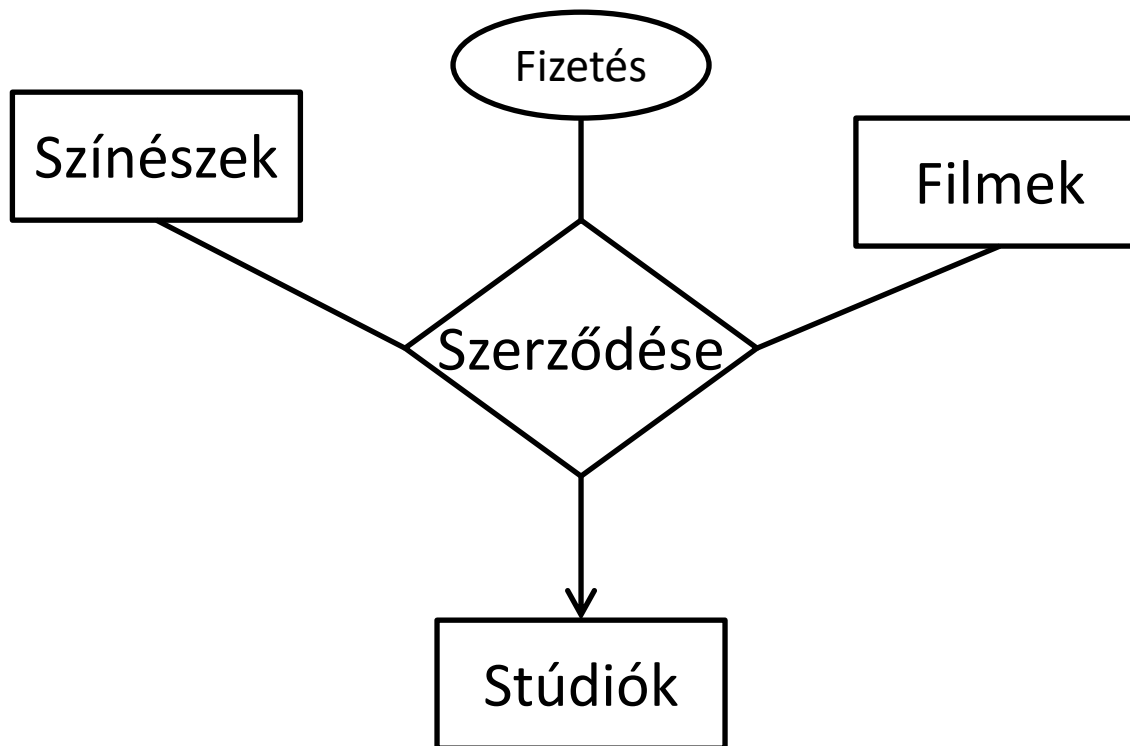


Többágú kapcsolat átírása



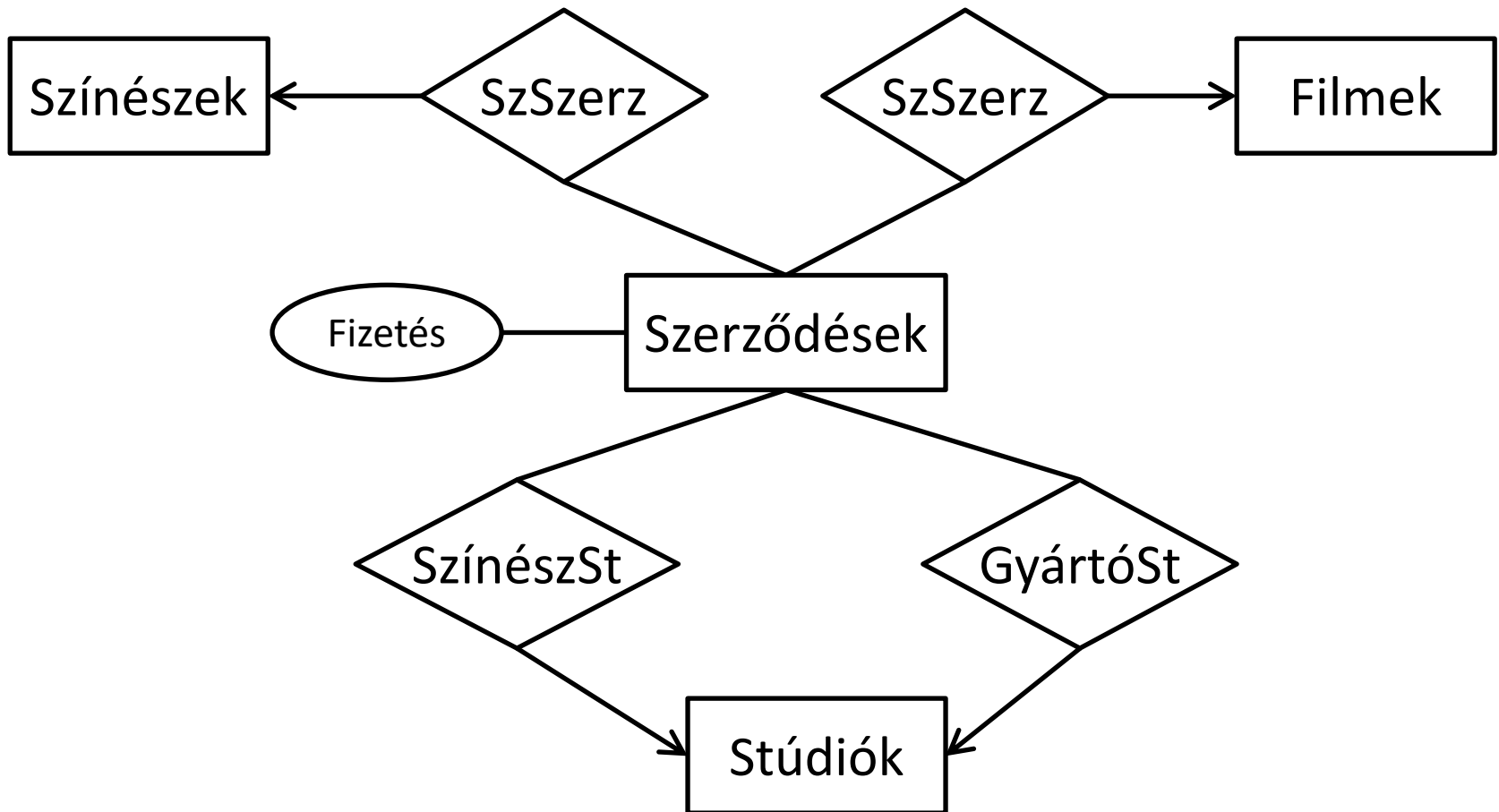
Többágú kapcsolat átírása

- 1.példa:

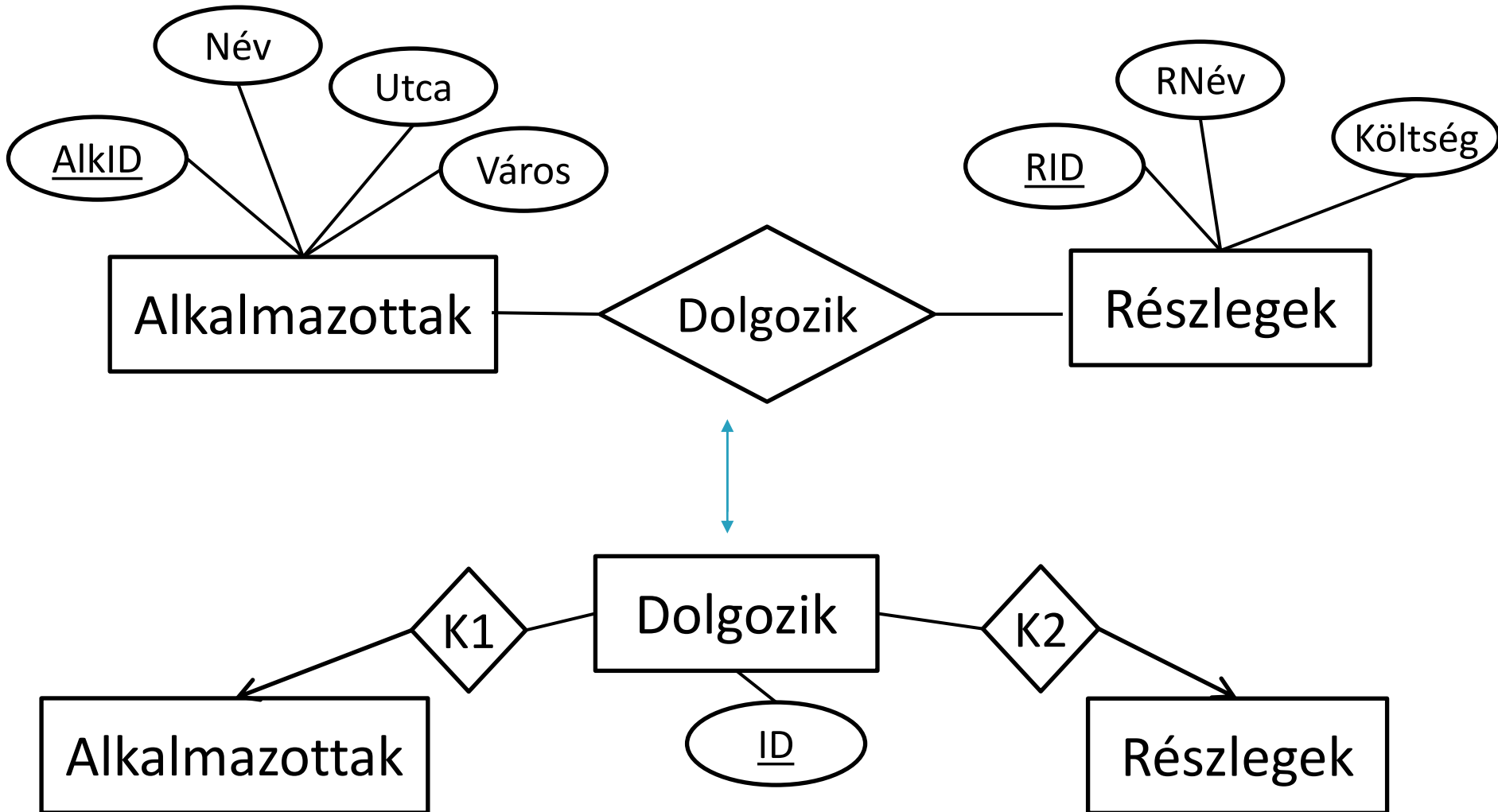


Többágú kapcsolat átírása

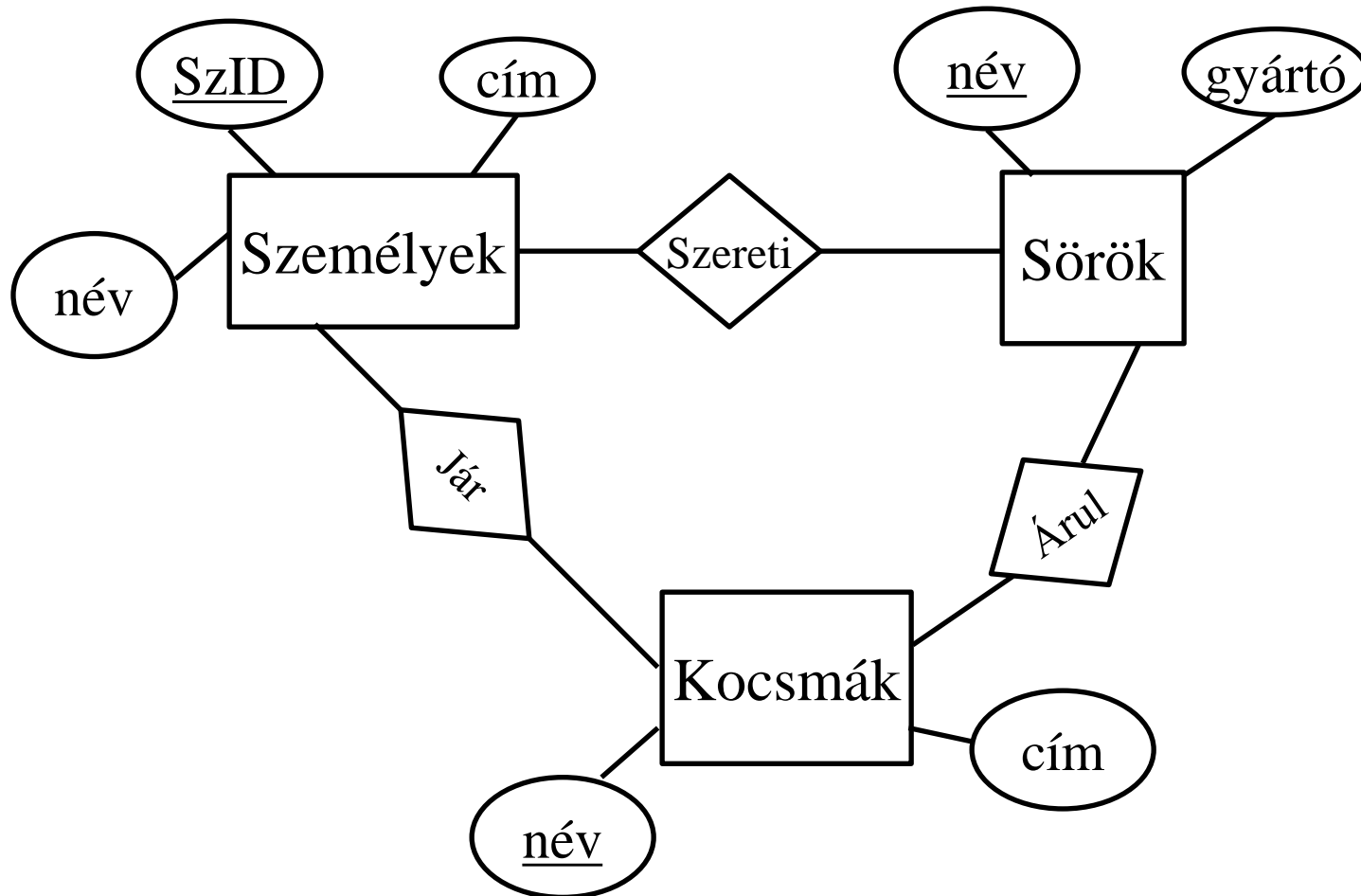
- 1.példa: 1 egyedhalmaz + 2 bináris kapcsolat segítségével



Bináris m:n kapcsolat ábrázolása



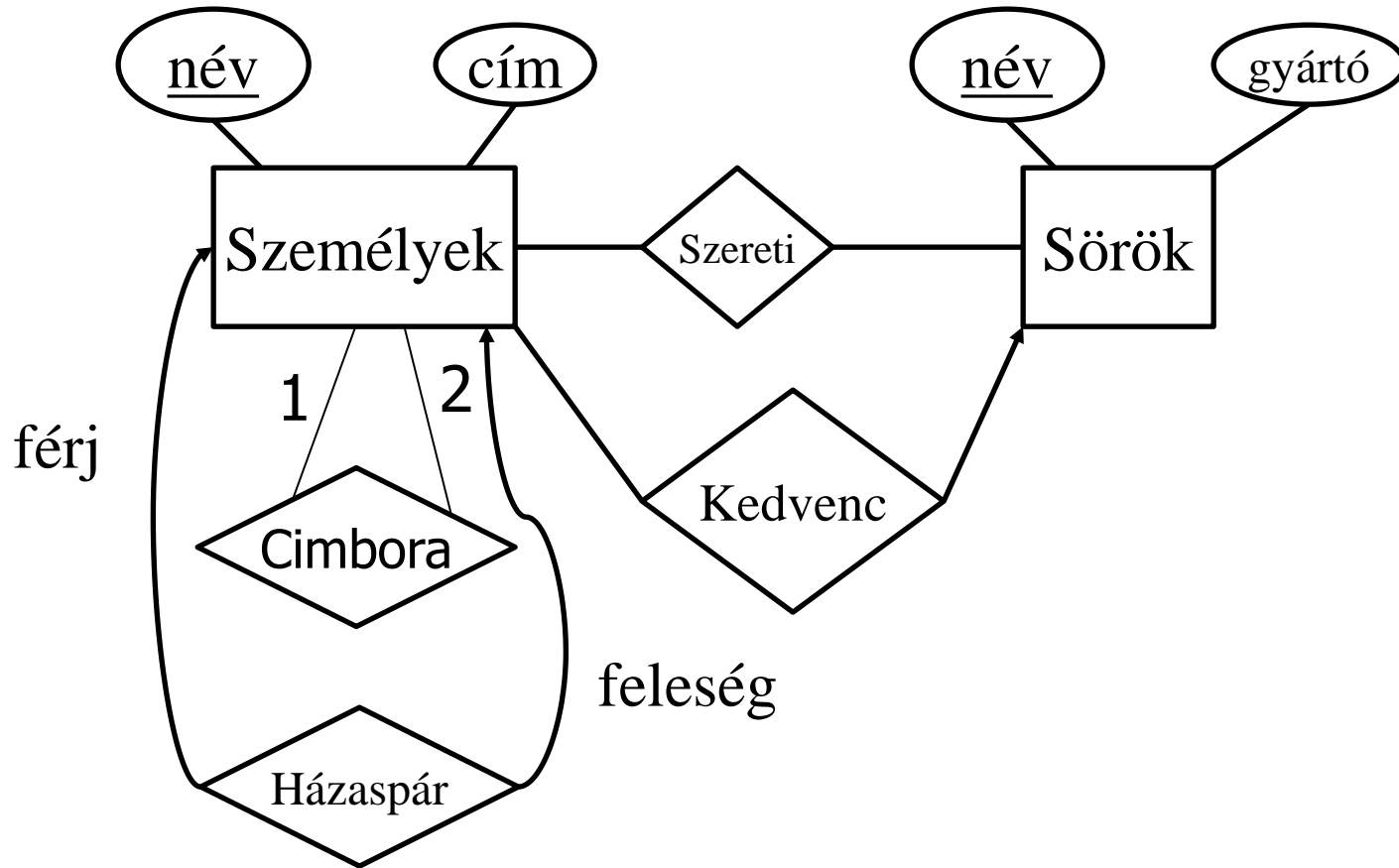
További példák



Az ilyen jellegű ábrázolásmóddal bánjunk elővigyázatosan – redundáns ábrázolás.

Példa – helyes, nem redundáns

További példák



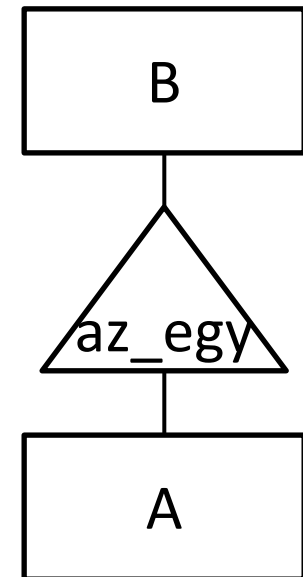
„Kapcsolat-különlegességek”

„az_egy” kapcsolat

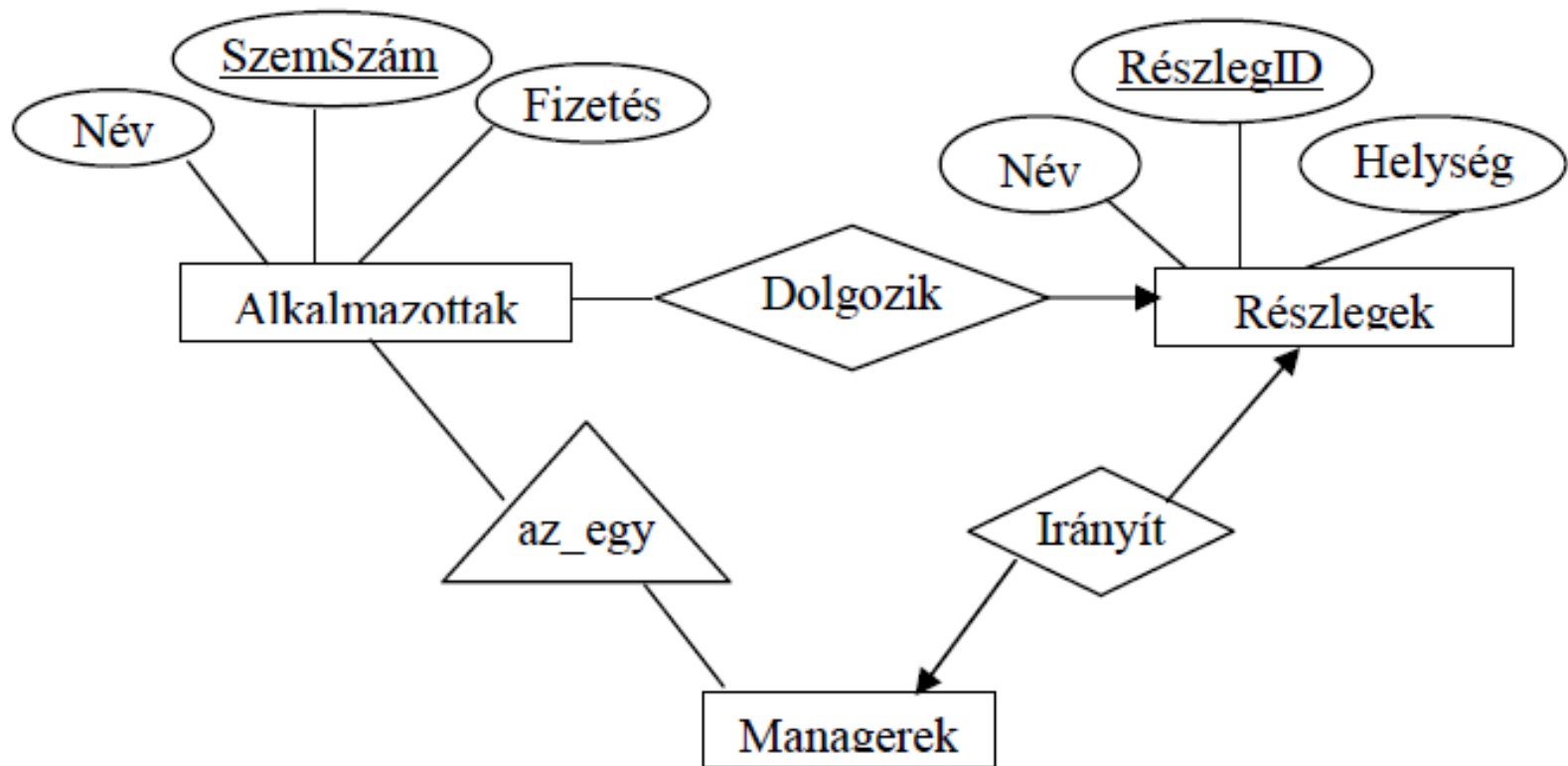
„Kapcsolat-különlegességek”

„az_egy” kapcsolat

- Öröklődési kapcsolat („az egy”, ISA).
- **A** „az_egy” **B** (**A** "is_a" **B**), ha a **B** egyedhalmaz az **A** egyedhalmaz egy **általánosítása**, másképp: **A** egy speciális **B**.
- A speciális egyed részhalmaz (**A**) az ősének (**B**) minden kapcsolatát és attribútumát örökli \leftrightarrow **B**-nek a kulcsa lesz **A**-nak a kulcsa is.
- *Más megnevezés:*
A-alosztály, **B**-(szuper)osztály

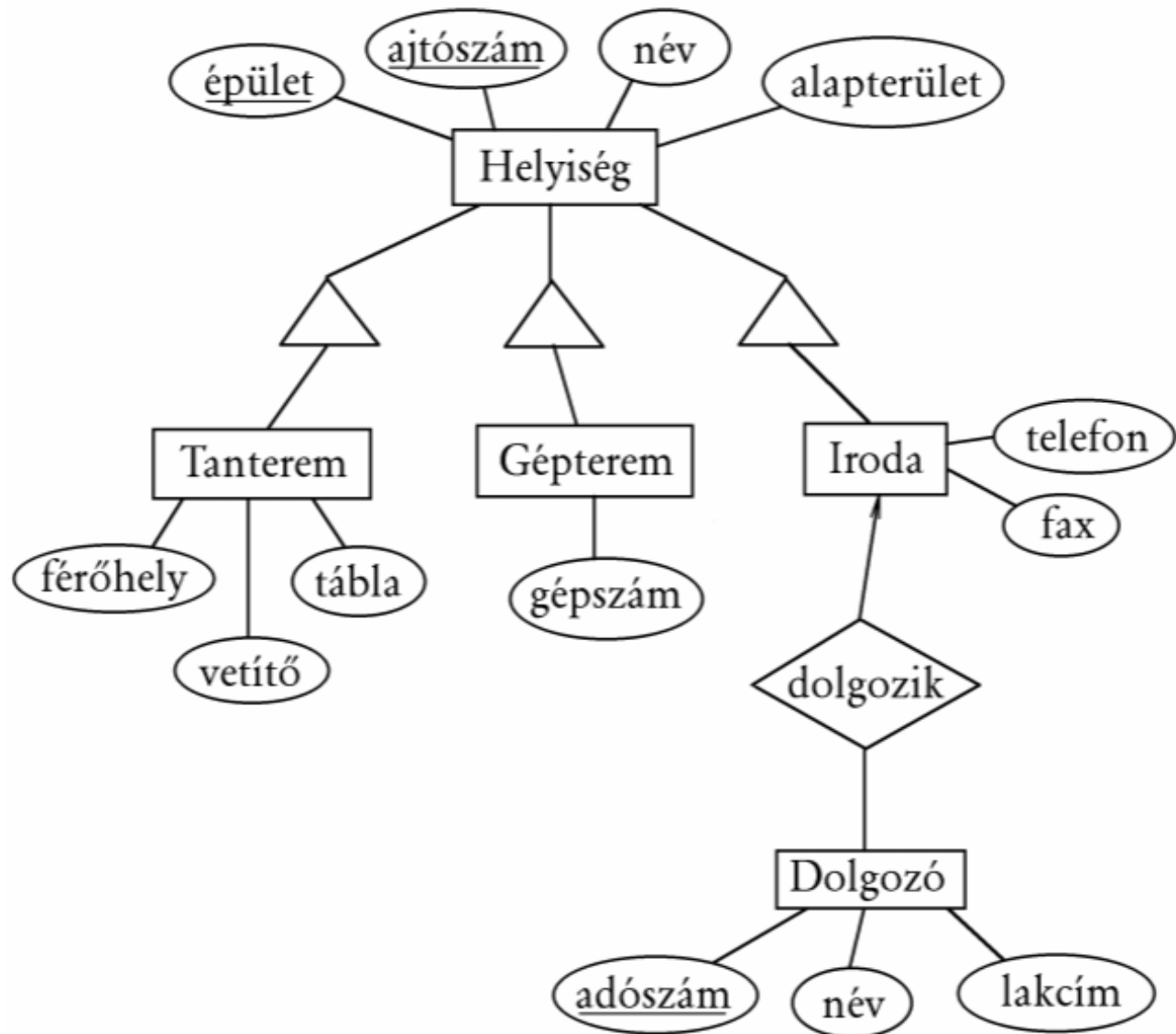


Példa „az_egy” kapcsolatra



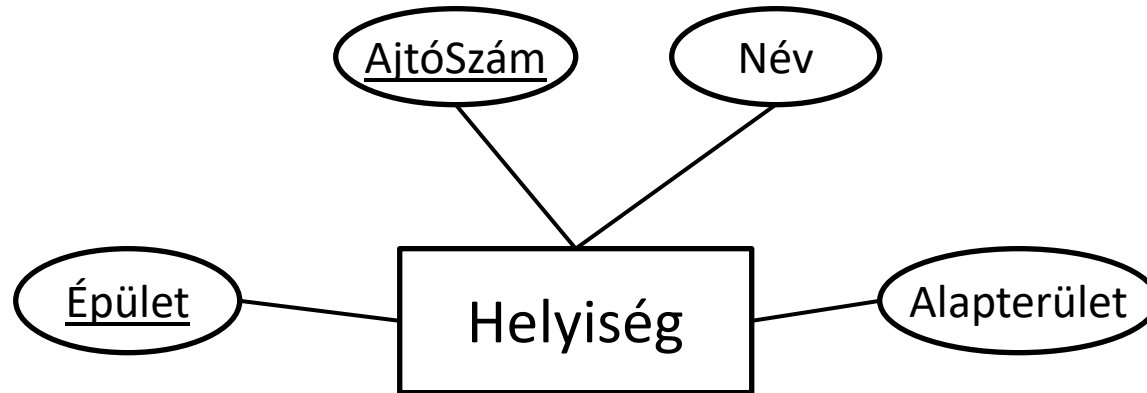
- manager „az_egy” alkalmazott

Újabb példa „az_egy” kapcsolatra



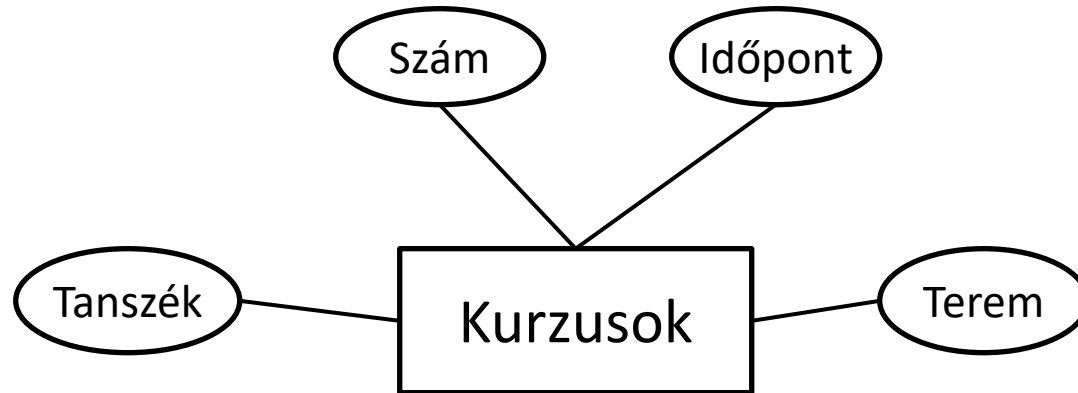
„Attribútum-különlegességek”

- **Többtényezős kulcs:** több attribútumból álló kulcs
- 1. Példa:



„Attribútum-különlegességek”

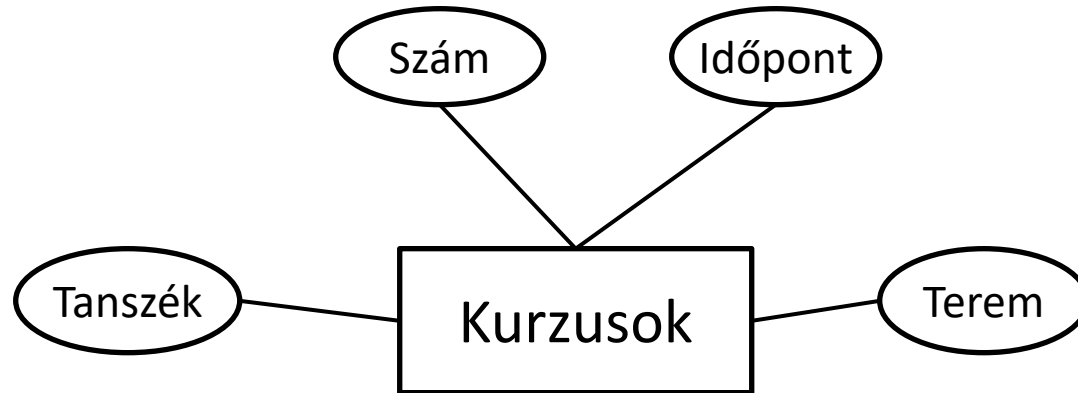
- **Többtényezős kulcs:** több attribútumból álló kulcs
- 2. példa:



- *Kulcsok (kulcsjelöltek)?*

„Attribútum-különlegességek”

- **Többtényezős kulcs:** több attribútumból álló kulcs
- 2. példa:



- *Kulcsok (kulcsjelöltek)?*
- Ritkán használjuk, helyette: „*több a többhöz*” kapcsolat javasolt.

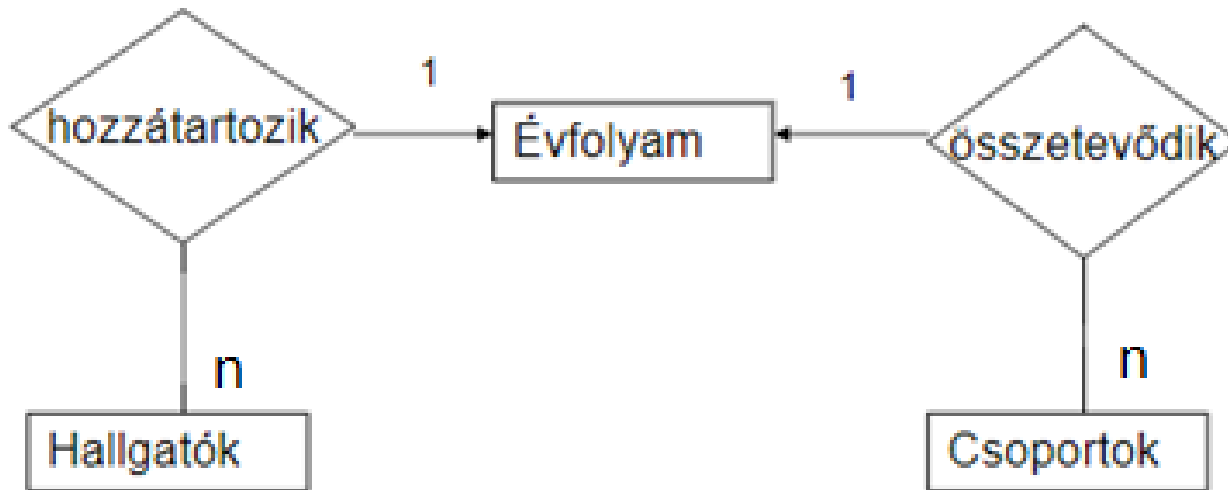
Megszorítások

- **Értelmezéstartomány-megszorítások:** egy attribútum az értékeit a megadott értékhalmból vagy értéktartományból veheti fel (lsd. DDL-ben: CHECK)
- **Egyértékűségi megszorítások:** Abban az esetben, ha az attribútum értéke nem NULL (*vagyis létezik*), akkor nem létezik a halmaznak 2 eleme, amelyiken megegyeznének az értékei (lsd. DDL-ben: UNIQUE).
- **Általános megszorítások:** tetszőleges követelmények, amelyeket be kell tartani az adatbázisban.

Tervezési problémák I.

Legyező csapda (fan trap)

- Előfordulás: legalább két egy-a-többhöz (1:N) típusú kapcsolat köti össze az egyedhalmazt

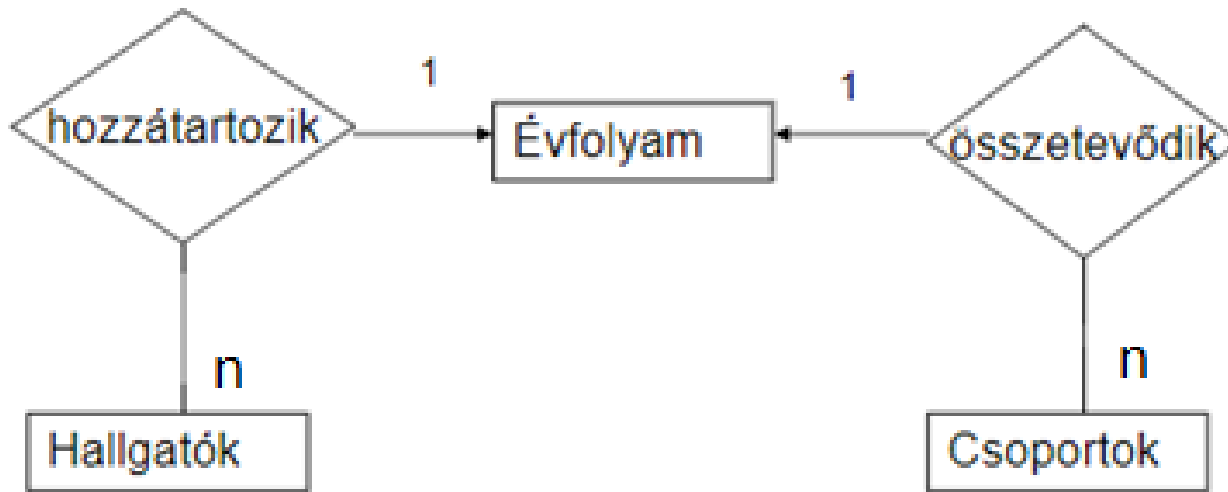


Probléma:

Tervezési problémák I.

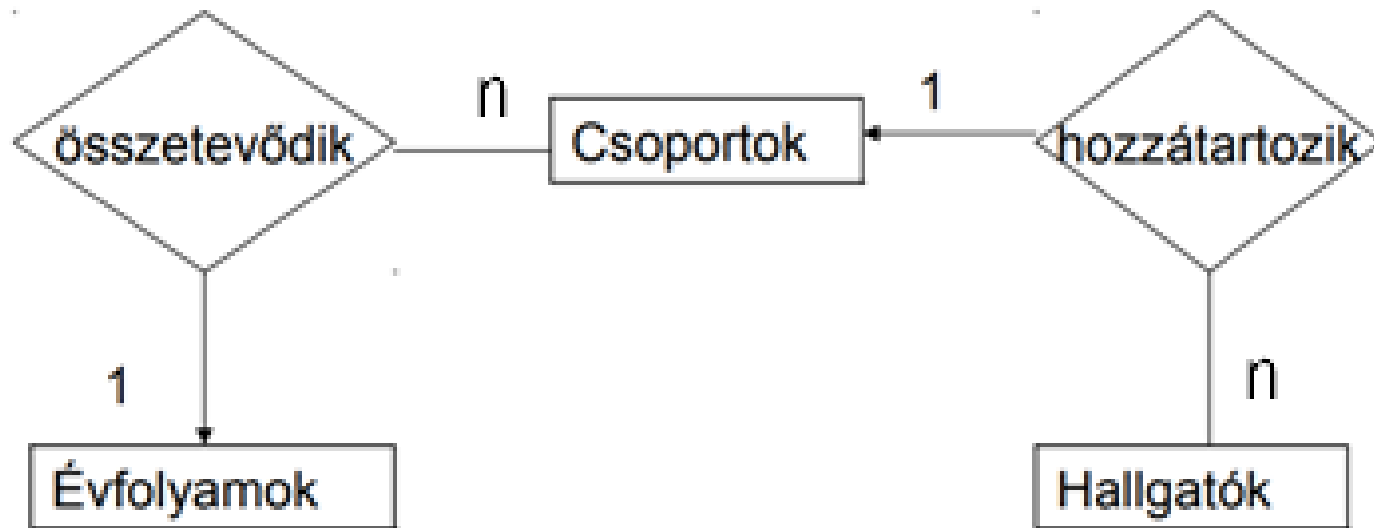
Legyező csapda (fan trap)

- Előfordulás: legalább két egy-a-többhöz (1:N) típusú kapcsolat köti össze az egyedhalmazt



Probléma: nem tudjuk megmondani, melyik hallgató melyik csoportba tartozik. – *Feloldás?*

Legyező csapda anomáliájának feloldása



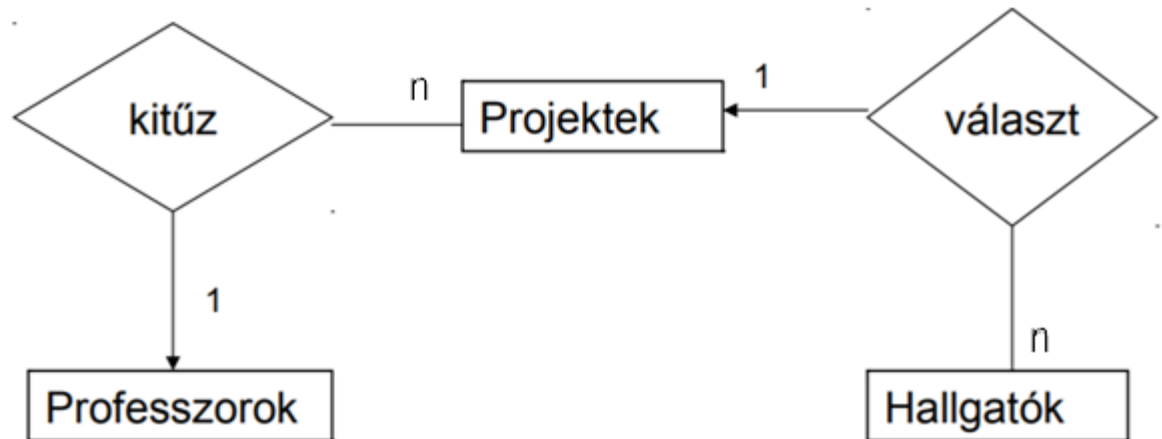
„Hármas hierarchia”:

Évfolyamok \leftarrow Csoportok \leftarrow Hallgatók

Tervezési problémák II.

Megszakítási csapda (chasm trap)

- Előfordulás: ha egy kapcsolat létezése javasolt az egyedhalmazok között, viszont az útvonal nem létezik bizonyos egyedelőfordulások között.

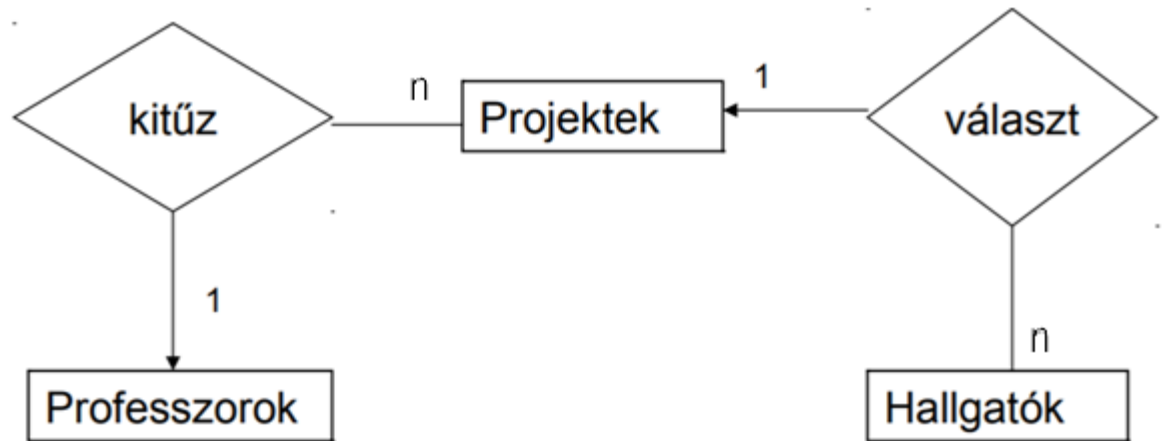


■ Probléma:

Tervezési problémák II.

Megszakítási csapda (chasm trap)

- Előfordulás: ha egy kapcsolat létezése javasolt az egyedhalmazok között, viszont az útvonal nem létezik bizonyos egyedelőfordulások között.



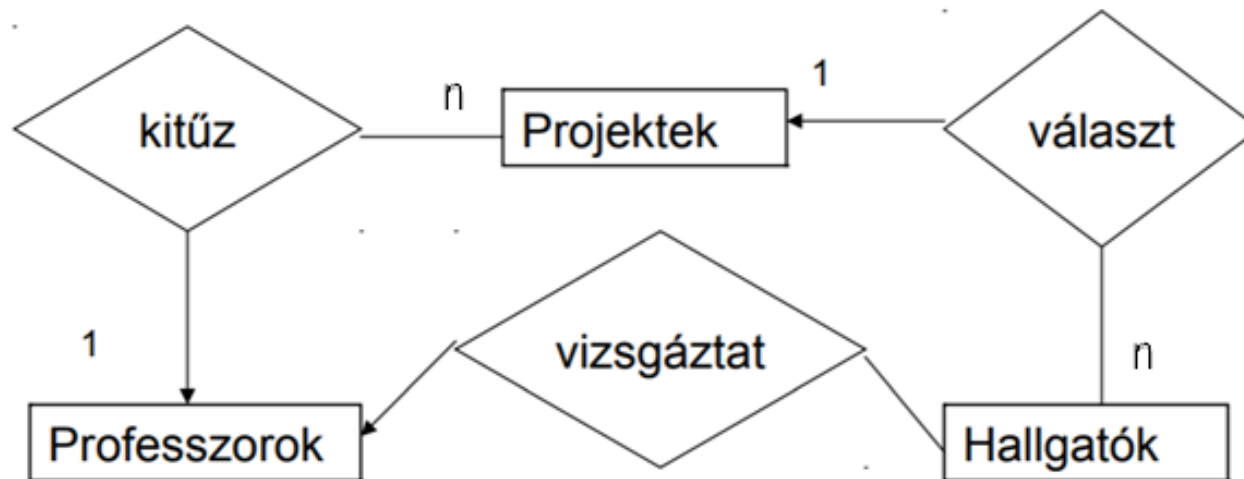
■ Probléma:

Ha egy hallgató
nem választott

egy projektet sem (másfajta vizsgázási módot választott),
nem tudjuk, hogy melyik professzornál kellene vizsgázzon.

Megszakítási csapda anomáliájának feloldása

- Új kapcsolat bevezetése



Tervezési alapelvek

- **Valósághű modellezés:** megfelelő tulajdonságok tartozzanak az egyedhalmazokhoz

Tervezési alapelvek

- **Valósághű modellezés**
- **Redundancia elkerülése:** ugyanazt az információt ne jelenítsük meg több helyen

Tervezési alapelvek

- **Valóságű modellezés**
- **Redundancia elkerülése:** ugyanazt az információt ne jelenítsük meg több helyen

pl. Studkönyv(NrMatr.,lakcím,tárgy,dátum,jegy) - rossz séma, mert a lakcím annyiszor ismétlődik, ahány vizsgajegye van a diáknak → helyette 2 séma:

Hallgatók(NrMatr.,lakcím)

Vizsgajegyek(NrMatr.,tárgy,dátum,jegy).

Tervezési alapelvek

Redundancia elkerülésének fontossága

- **Helygazdálkodás:** nem a legfontosabb szempont, de szükséges mérlegelni
- **Konzisztencia**
- **Egyszerűség:** a redundancia bonyolítja a modellt
Pl. fölöslegesen ne vegyünk fel egyedhalmazokat:
pl. naptár(év,hónap,nap) helyett – dátum tulajdonság

Tervezési alapelvek

Megfelelő típusú elemek kiválasztása

Alapkérdés: *Mit reprezentálunk attribútumként és mit kapcsolatként vagy épp egyedhalmazként?*

Példák:

pl. Vizsgajegy(jegy) helyett – jegy tulajdonság használata

Tervezési alapelvek

- **Megfelelő típusú elemek kiválasztása**

Alapkérdés: *Mit reprezentálunk attribútumként és mit kapcsolatként vagy épp egyedhalmazként?*

pl. Polgármesteri hivatal – házasságok nyilvántartása

Az adatbázis tervezésének 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. Adatbázisterv megvalósítása SQL-ben (DDL segítségével)

Relációs adatmodell

Relációs adatmodell

név	ország
Soproni	Magyar
Kozel	Cseh
Dreher	Német

Sörök (név, ország)

- E. F. Codd (1970) vezette be a “*A Relational Model of Data for Large Shared Data Banks*” című cikkben ([link](#)).
 - Javaslat: az adatokat táblázatokban, relációkban tároljuk.
 - Az elméletére alapozva jött létre a **relációs adatmodell**, és erre épülve jöttek létre a relációs adatmodellen alapuló relációs ABKR-k (röv. RDBMS-relational DBMS).
- Gyakorlati alkalmazása az 1980-as években vált általánossá.
- Jelenleg a legszélesebb körben használatos adatmodell.

Relációs adatmodell

név	ország
Soproni	Magyar
Kozel	Cseh
Dreher	Német

Sörök (név, ország)

- Lényege: az egyedeket, tulajdonságokat és kapcsolatokat egyaránt táblázatok, ún. adattáblák (relációk) segítségével adja meg.
- **Értékorientált:** reláción végzett műveletek eredménye - reláció + azonosító megadása szükséges két sor megkülönböztetése érdekében (nincs ObjectID – lsd. obj.or. adatmodell)
- Szabványos leíró/lekérdezőnyelvük: SQL.

Relációs adatmodell előnyei

- Egy egyszerű és könnyen megérthető **strukturális részt** tartalmaz (táblázatos forma).
- Nagyfokú fizikai és logikai adatfüggetlenség biztosítása (fogalmi-logikai-fizikai szint teljes szétválasztása)
- Elméleti megalapozottság, több absztrakt kezelőnyelv megléte (például *relációs algebra* – az SQL automatikus és **hatékony lekérdezés optimalizálásának** alapja).
- Műveleti része egyszerű kezelői felület, szabvány: SQL.

Relációs adatmodell értelmezése

- Legyenek: $A_1, A_2 \dots, A_n$ attribútumok.
- $\forall A_i : D_i$ - A_i értékeinek tartománya (doméniuma).
 - D_i lehet: egész számok halmaza, karaktersorok halmaza stb.
- **R reláció** a $D_1, D_2 \dots, D_n$ halmazokon, ha
$$R \subseteq D_1 \times D_2 \times \dots \times D_n \text{ (Descartes-szorzat)}$$
- Jelölés: $R(A_1, A_2, \dots, A_n)$ (*)
- **Relációséma**: reláció neve, attribútumok halmazának + további szerkezeti információk (pl. kulcsok-függések, megszorítások stb.) együttese \Rightarrow (*) tulajdonképp relációséma.
- ***Mi akkor a reláció?***

Relációs adatmodell értelmezése

- **Reláció:** a konkrét, adatokkal feltöltött, sémára illeszkedő táblázat (a táblázat soraiban tárolt adatokkal együtt).
- Továbbiakban: **relációsémára relációként fogunk hivatkozni.**

Egy adatbázis
relációsémáinak
összessége: **relációs
adatbázisséma.**

<u>A1</u>	A2	A3	...	An
a1	a2	a3		an
b1	b2	a3		cn
a1	c3	b3		bn
.	.	.		.
x1	v2	d3		wn

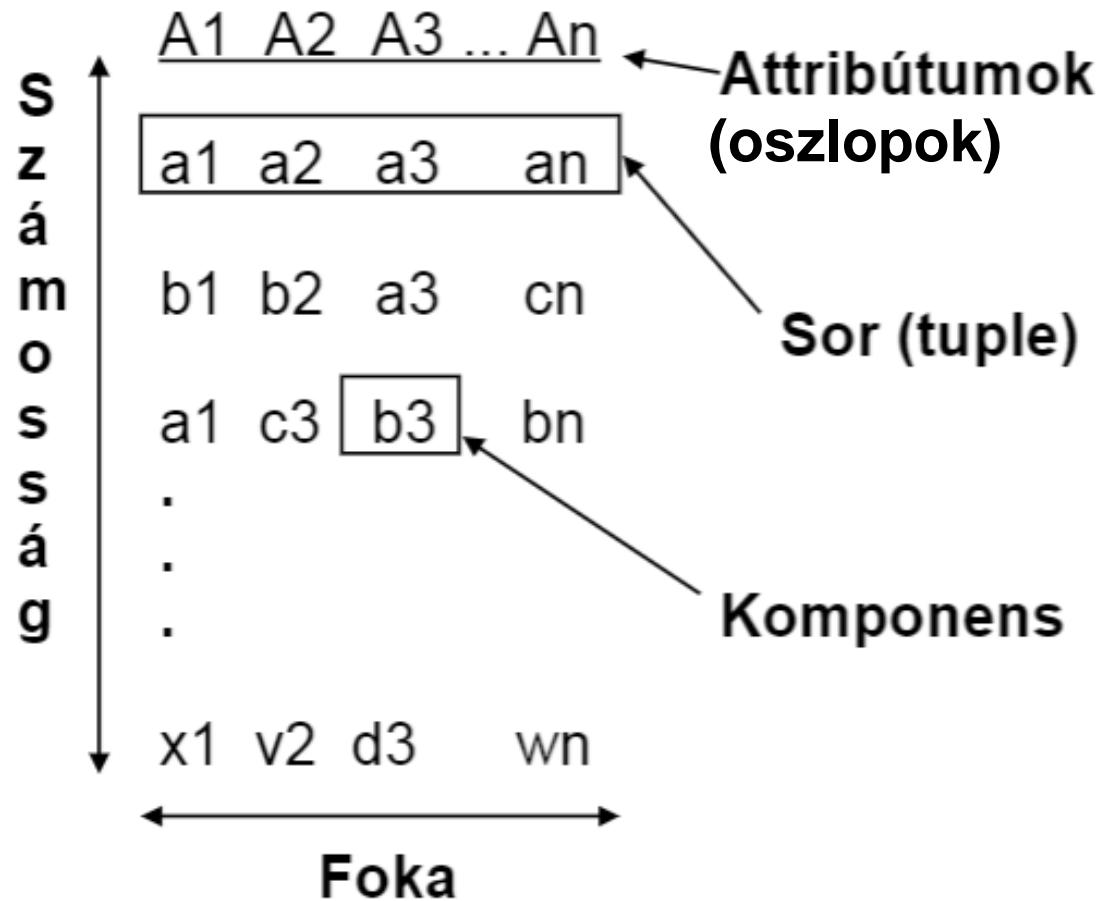
Attribútumok
(oszlopok)

Sor (tuple)

Komponens

Relációs adatmodell értelmezése

- Reláció számossága \leftrightarrow sorok száma
- Reláció foka \leftrightarrow attribútumok száma (itt: n)
- Reláció előfordulás: aktuális sorok halmaza egy relációsémára vonatkozóan.



Példa relációs adatmodellre

- **Egy reláció sémája:** Sörök (név, ország)
 - Sörök reláció foka: 2, számossága: 3
- **Adatbázis sémája:** Sörök (név, ország), Sörözők (név, város, tulaj), Felszolgál(sör, bár, ár)



Példa relációs adatmodellre

■ Relációsémák más megadási módja:

Sörök (név:string, ország:string)

Sörözők (név:string, város:string, tulaj:string)

Felszolgál(sör:string, bár:string, ár:int)



Relációs adatmodell tulajdonságai

- A táblának és a táblák oszlopainak (attribútumainak) egyértelmű neve, helye, sorszáma van.
 - Az oszlopokra a nevükkel hivatkozunk.
 - **Két attribútumnak nem lehet ugyanaz a neve egy táblán belül.** (1.tulajd.)
- Tábla **oszlopainak sorrendje - lényegtelen.** (2.tulajd.)
- Reláció **sorainak sorrendje lényegtelen** (a sorokat halmazként kezeli). (3.tulajd.) - *Tárolásnál: fontos!*
- **A tábla nem tartalmazhat két teljesen azonos sort.** (4.tulajd.)

Relációs adatmodell tulajdonságai

- A táblának és a táblák oszlopainak (attribútumainak) egyértelmű neve, helye, sorszáma van.
 - Az oszlopokra a nevükkel hivatkozunk.
 - **Két attribútumnak nem lehet ugyanaz a neve egy táblán belül.** (1.tulajd.)
- Tábla **oszlopainak sorrendje - lényegtelen.** (2.tulajd.)
- Reláció **sorainak sorrendje lényegtelen** (a sorokat halmazként kezeli). (3.tulajd.) - *Tárolásnál: fontos!*
- **A tábla nem tartalmazhat két teljesen azonos sort.** (4.tulajd.)
 - **Kulcs értelmezése** (5.tulajd.)

Kulcsok

Értelmezés: Legyen $A = \{A_1, A_2, \dots, A_n\}$ az R reláció attribútumhalmaza. A $K \subseteq A$ halmaz az R **reláció kulcsa**, ha:

- 1) a K attribútumain felvett értékek egyértelműen meghatározzák a reláció elemeit;
- 2) K *minimális* \Leftrightarrow K -nak nincs olyan valódi részhalmaza, amelyre ugyanez teljesül.

Pl. Diákok (BeiktSzám, Név, Cím, Telefonszám)

- BeiktSzám – kulcs.

- **Szuperkulcs** (*superkey*) – azon attribútumhalmaz, amelyre csak az 1) tulajdonság teljesül.

- Pl. {BeiktSzám, Név} – szuperkulcs.

Kulcsok (folyt.)

- **Kulcsjelölt** (*candidate key*) – **minimális(!) attribútumhalmaz**
– értéke(i) egyedi(ek) minden egyes előfordulásra nézve
(2). *tulajd. is teljesül*
 - \exists több kulcsjelölt is - *példa?*

Kulcsok (folyt.)

- **Kulcsjelölt** (*candidate key*)

- \exists több kulcsjelölt is

pl. Cégek (CégNév, BejegyzésiSzám, CUI, CIF, AlakulásiÉv, Székhely)

- CégNév, BejegyzésiSzám, CUI, CIF - kulcsjelöltek.

- Egyet kiválasztunk közülük \leftrightarrow **elsődleges kulcs** (*primary key*)

- **Többi** kulcsjelölt (*alternative key*) esetén: egyértékűségi megszorítás (*lsd. SQL-megszorítások*)

- Ha a kulcs csak egy attribútumot tartalmaz \leftrightarrow **egyszerű kulcs**.
Ellenkező esetben: \leftrightarrow **összetett kulcs**.

- 4. tulajdonság \Rightarrow ***mindig kell legyen elsődleges kulcs***, ha más nem: **a teljes sor mindig egyedi**.

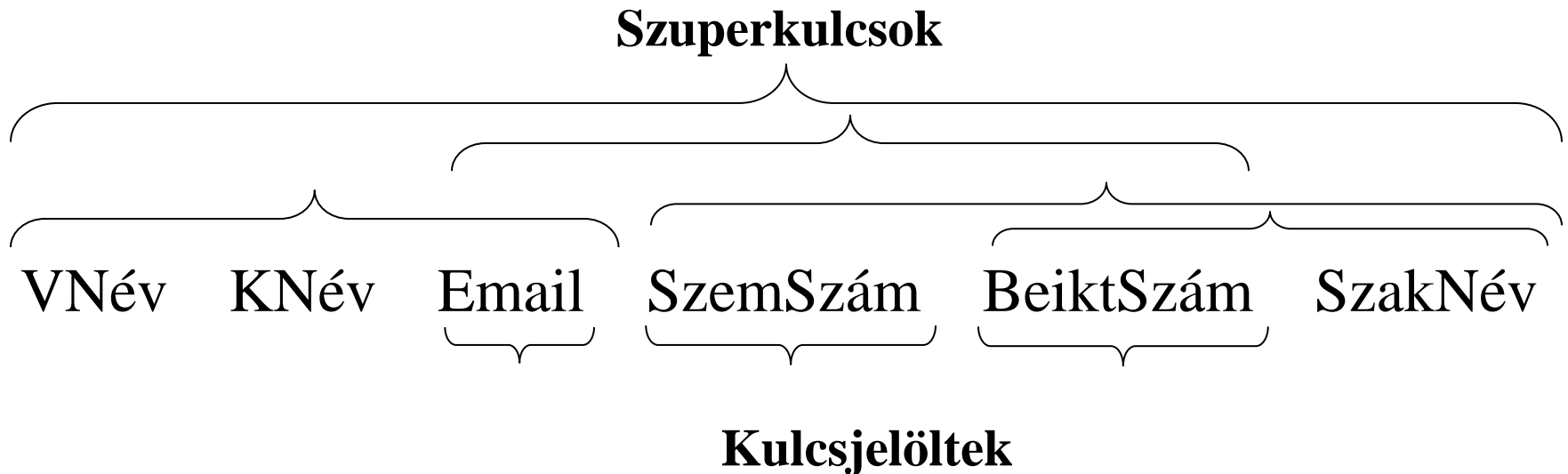
- **Elsődleges kulcs értéke soha *nem lehet null vagy üres***.

Kulcsok - példa

Diákok (VNév, KNév, Email, SzemSzám, BeiktSzám, SzakNév)

Kulcsok - példa

Diákok (VNév, KNév, Email, SzemSzám, BeiktSzám, SzakNév)



Külső kulcs (foreign key)

Definíció: **Külső kulcs** egy **KK** attribútum(halmaz) egy **R_f** relációból ú.h.:

- értékeinek halmaza megegyezik egy **R_a** reláció elsődleges kulcsának az értékalmazásával;
- feladata: az **R_f** és **R_a** közötti kapcsolat modellezése.

- **R_f** - reláció, mely **hivatkozik**; **R_a** – reláció, **amelyre hivatkozik**.

Külső kulcs (foreign key)

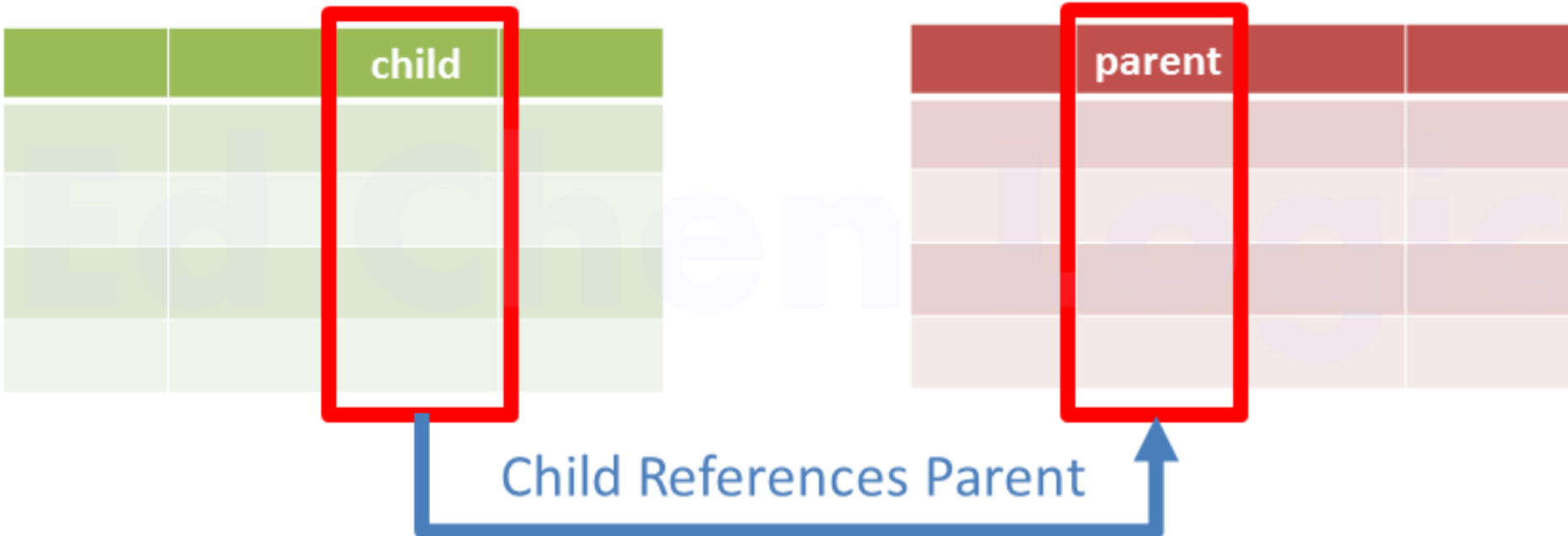
R_f - reláció, mely **hivatkozik**

R_a – reláció, **amelyre** **hivatkozik**.

Más megnevezés: R_a – **apa** reláció, R_f – **fiú** reláció

Child Table

Parent Table



Példa külső kulcsra

↓ Részlegek - *apa* reláció

<i>RészlegID</i>	<i>RNév</i>
1	Tervezés
2	Könyvelés
9	Beszerzés

Alkalmazottak – *fiú* reláció

<i>SzemSzá</i> m	<i>Név</i>	<i>RészlegID</i>	<i>Fizetés</i> (<i>euró</i>)
111111	Nagy Éva	2	300
222222	Kiss Csaba	9	400
456777	Szabó János	9	900
234555	Szilágyi Pál	2	700
123444	Vincze Ildikó	1	800
333333	Kovács István	2	500

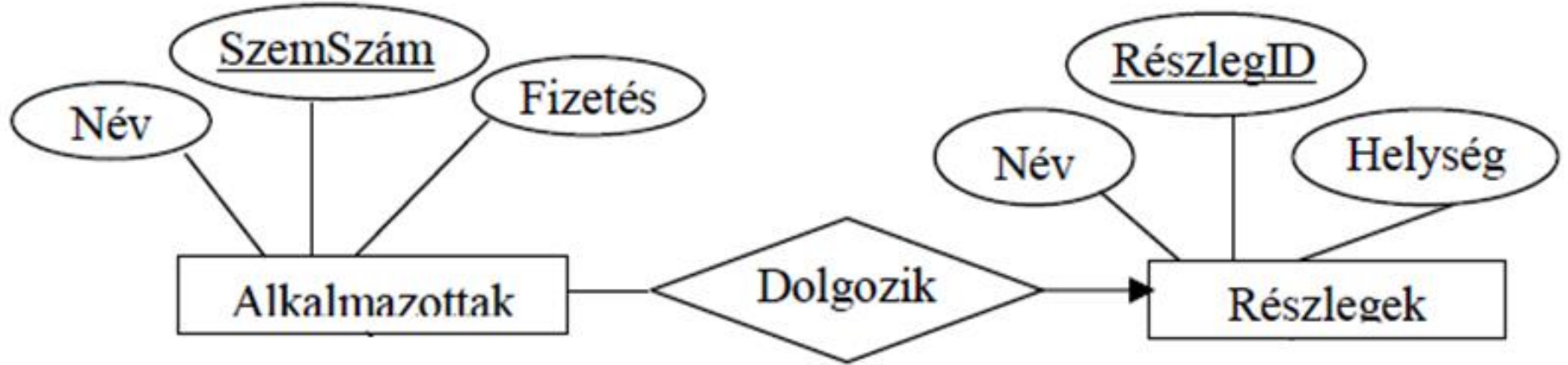
Ábrázolás
E/K diagrammal?

Relációk sémája:

Részlegek (RészlegID, RNév)

Alkalmazottak (SzemSzám, Név, *RészlegID*, Fizetés)

Modellezés E-K diagrammal



Példa külső kulcsra

Sörök

név	ország
Soproni	Magyar
Kozel	Cseh
Dreher	Német

Sörözők

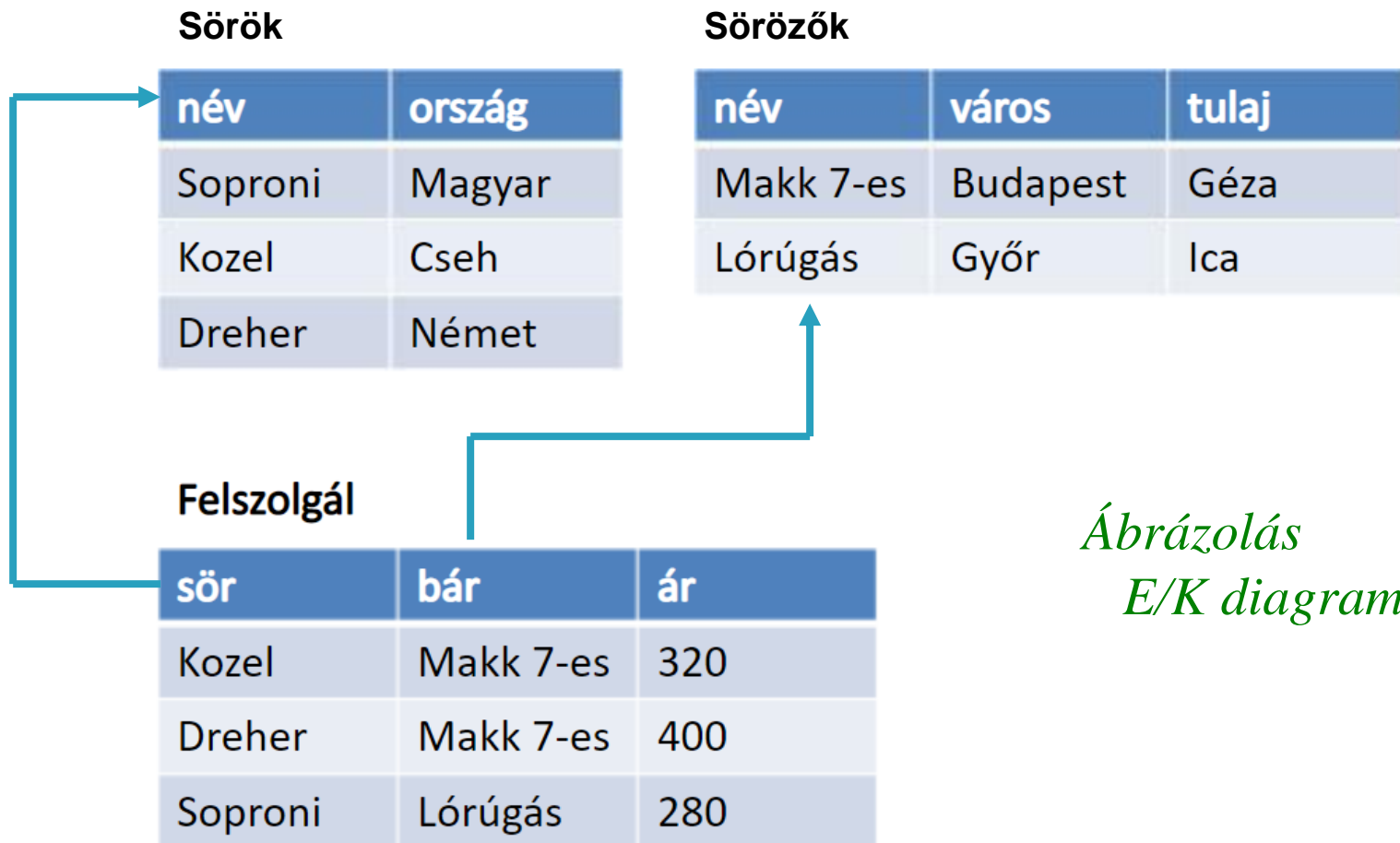
név	város	tulaj
Makk 7-es	Budapest	Géza
Lórúgás	Győr	Ica

Felhasználók

sör	bár	ár
Kozel	Makk 7-es	320
Dreher	Makk 7-es	400
Soproni	Lórúgás	280



Modellezés E-K diagrammal



*Ábrázolás
E/K diagrammal?*

Külső kulcs

- Egy relációban több idegen (külső) kulcs is szerepelhet + ugyanazon reláció kulcsára többször is hivatkozhatunk ugyanazon reláción belül.

Példa:

Külső kulcs

- Egy relációban több idegen (külső) kulcs is szerepelhet + ugyanazon reláció kulcsára többször is hivatkozhatunk ugyanazon reláción belül.

Példa:

Helységek (HelységID, HelységNév)

Személyek (SzemSzám, Név, *LakhelyID*, *SzülHelyID*)



- *Lásd még: hivatkozási épség megszorítás*

Az egyed-kapcsolat diagramok átírása relációs modellé

Relációs adatbázisséma meghatározása

- *(Ism.)* **Relációs adatbázisséma:** az adatbázist alkotó relációk sémájának az összessége.
- **Relációs adatbázis:** a relációkban tárolt konkrét értékek alkotják.
- Relációk „helyességének” biztosítása az adatbázisban - egy metodológia szükséges.

Több lehetőség:

- *Egyed-kapcsolat adatmodell* - átírása relációsémákká
- *Normalizálás* - egy létező relációs adatbázissémát normalizálás segítségével normálformára hozzuk.
- Object Definition Language segítségével megtervezett objektum-orientált adatbázis-szerkezetet átírunk relációsémákká
- UML diagram - átalakítása relációsémákká

Az egyed-kapcsolat diagramok relációs modellé való átírásának lépései

3 lépés:

1. *Egyedhalmaz* \rightarrow *reláció*

- Egyedhalmaz attribútumai a reláció attribútumai lesznek.

2. *Kapcsolat* \rightarrow *reláció*

- A reláció attribútumai: a kapcsolatban résztvevő egyedhalmazok kulcsainak uniója + kapcsolat attribútumai (sok esetben átnevezés szükséges lehet).

Az egyed-kapcsolat diagramok átírása relációs modellé

2. *Kapcsolat* \rightarrow *reláció*

- Legyen K **bináris** kapcsolat E_1 és E_2 egyedhalmazok között
 - K_1, K_2 – E_1 -nek, illetve E_2 -nek a kulcsa
- **1:1** típusú kapcsolatok esetén a kapcsolatnak megfelelő reláció (K) kulcsjelöltje: ...
- **1:N** típusú kapcsolat esetén (E_2 - itt az n) K kulcsjelöltje: ...
- Ha **N:M** típusú kapcsolat áll fenn, K kulcsjelöltje: összetett kulcs ...
- Az E/K diagram „**A az_egy B**” specializáló kapcsolatai esetén: ...

Az egyed-kapcsolat diagramok átírása relációs modellé

2. *Kapcsolat* \rightarrow *reláció*

- Legyen K **bináris** kapcsolat E_1 és E_2 egyedhalmazok között
 - K_1, K_2 – E_1 -nek, illetve E_2 -nek a kulcsa
- **1:1** típusú kapcsolatok esetén a kapcsolatnak megfelelő reláció (K) kulcsjelöltje: **K_1 vagy K_2** .
- **1:N** típusú kapcsolat esetén (E_2 - itt az N) K kulcsjelöltje: **K_2** .
- Ha **N:M** típusú kapcsolat áll fenn, K kulcsjelöltje: összetett kulcs - **K_1 és K_2 egyesítése**. *Fennállhat-e olyan eset, amikor (K_1, K_2) nem elegendő a sor azonosítására?*
- Az E/K diagram „**A az_egy B**” specializáló kapcsolataihoz *nem készítünk relációkat*. A örökli B attribútumait, B-nek a kulcsa lesz A-nak a kulcsa is.

Az egyed-kapcsolat diagramok relációs modellé való átírásának lépései

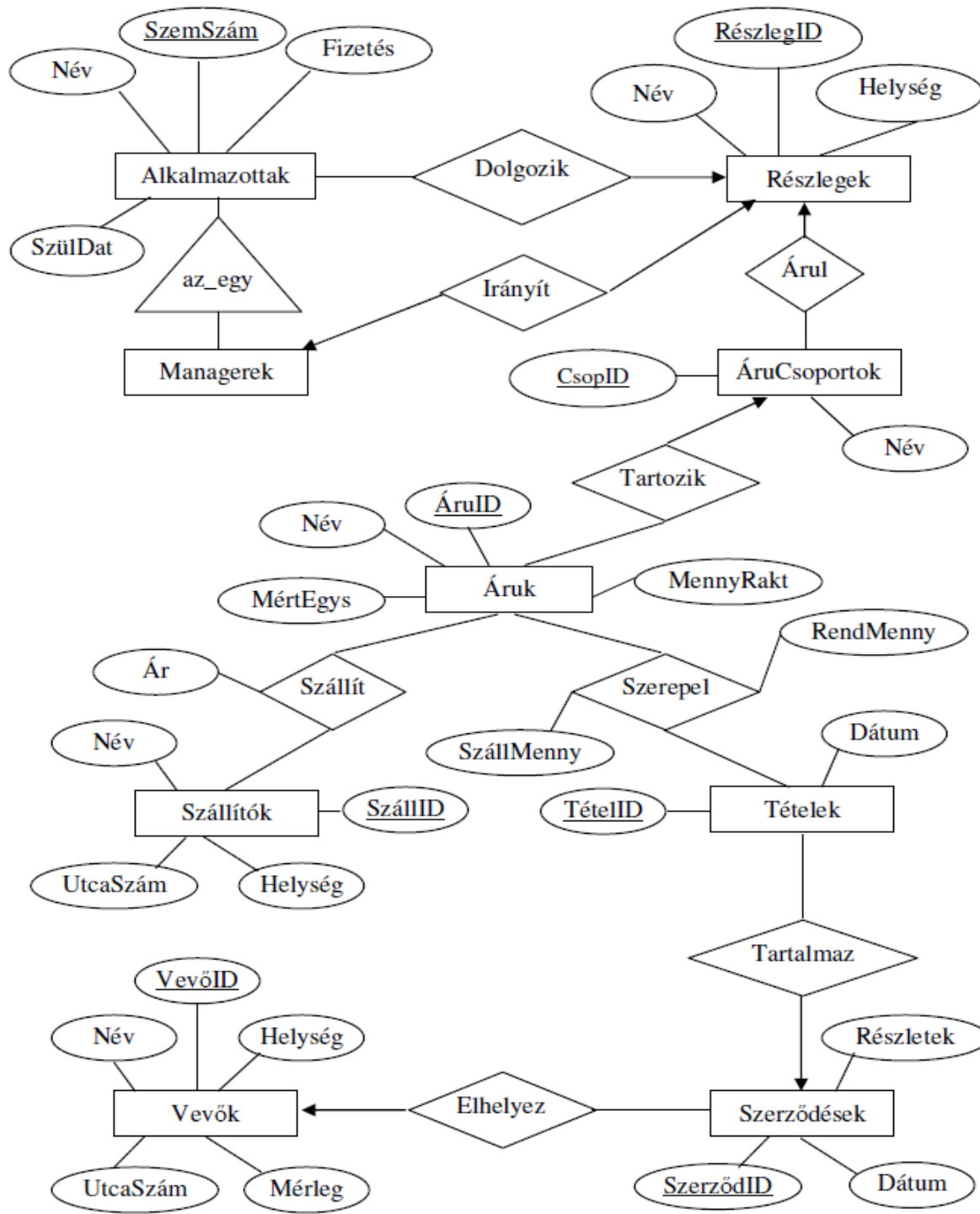
3. lépés: *Közös kulcsú relációk összevonása*

- Ha két vagy több relációnak van egy közös kulcsjelöltje (1. lépésből legtöbb egy reláció) → a két relációt/relációkat összevonjuk és helyettesítjük egy újabb relációval.
- **Előnyök:**
 - Helytakarékoság
 - Összetartozó adatok nincsenek szétdarabolva ↔ lekérdezések megválaszolási ideje csökkenthető

Példa

Nagykereskedő cég
egyszerűsített
adatbázisa

(leírásért lsd.
Varga Ibolya:
Adatbázisrendszerek)



Lépésenkénti átírás

I. Egyedhalmaz \rightarrow reláció

- Egyedhalmazok kulcsai \rightarrow relációk kulcsai
- A kulcsokat a relációs modell relációi esetében is aláhúzással jelöljük.

- (1) Alkalmazottak (SzemSzám, Név, Fizetés)
- (2) Managerek (SzemSzám)
- (3) Részlegek (RészlegID, Név, Helység)
- (4) Szállítók (SzállID, Név, Helység, UtcaSzám)
- (5) Árucsoportok (CsopID, Név)
- (6) Áruk (ÁruID, Név, MértEgys, MennyRakt)
- (7) Vevők (VevőID, Név, Helység, UtcaSzám, Mérleg)
- (8) Szerződések (SzerződID, Dátum, Részletek)
- (9) Tételek (TételID, Dátum)

Lépésenkénti átírás

II. Kapcsolat → reláció

- kulcsok → attribútumok

(10) Dolgozik(SzemSzám, RészlegID)

(11) Irányít (SzemSzám, RészlegID)

(12) Árul (CsopID, RészlegID)

(13) Tartozik (CsopID, ÁruID)

(14) Szállít (SzállID, ÁruID, Ár)

(15) Elhelyez (VevőID, SzerződID)

(16) Tartalmaz (SzerződID, TételID)

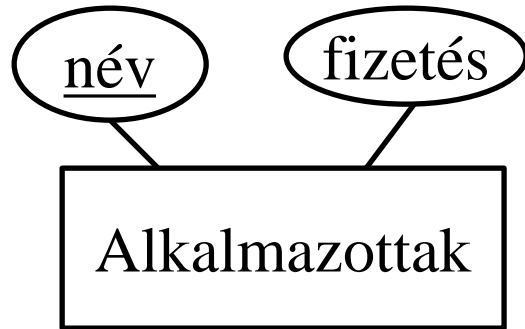
(17) Szerepel (TételID, ÁruID, RendMenny, SzállMenny)

Lépésenkénti átírás

III. Közös kulcsú relációk összevonása

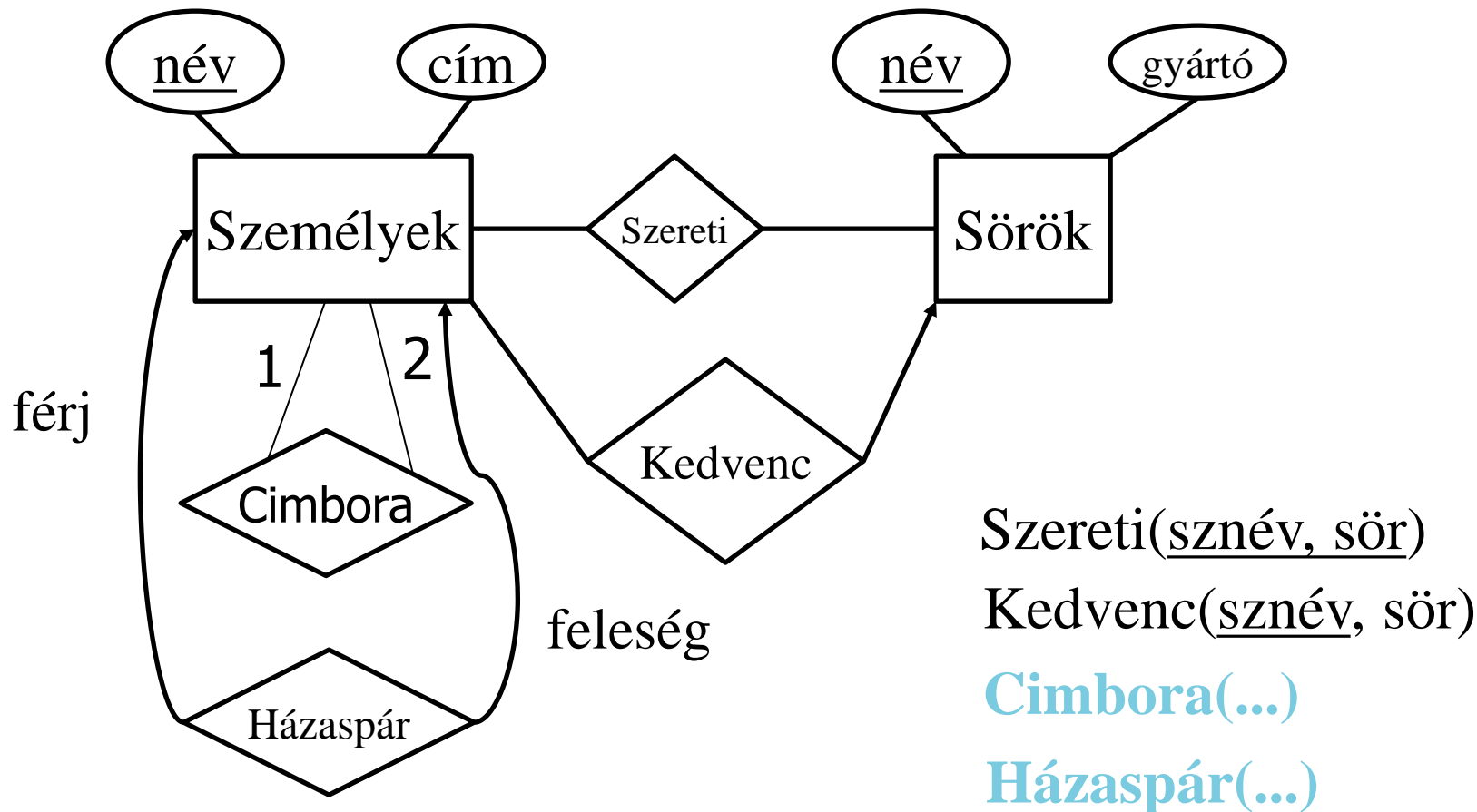
- (1) Alkalmazottak (SzemSzám, Név, Fizetés, *RészlegID*)
- (2) Managerek (SzemSzám)
- (3) Részlegek (RészlegID, Név, Helység, *ManSzemSzám*)
- (4) Szállítók (SzállID, Név, Helység, UtcaSzám)
- (5) Árucsoportok (CsopID, Név, *RészlegID*)
- (6) Áruk (ÁruID, Név, MértEgys, MennyRakt, *CsopID*)
- (7) Vevők (VevőID, Név, Helység, UtcaSzám, Mérleg)
- (8) Szerződések (SzerződID, Dátum, Részletek, *VevőID*)
- (9) Tételek (TételID, Dátum, *SzerződID*)
- (14) Szállít (SzállID, ÁruID, Ár)
- (17) Szerepel (TételID, ÁruID, RendMenny, SzállMenny)

Példa (Kapcsolat \rightarrow Reláció)

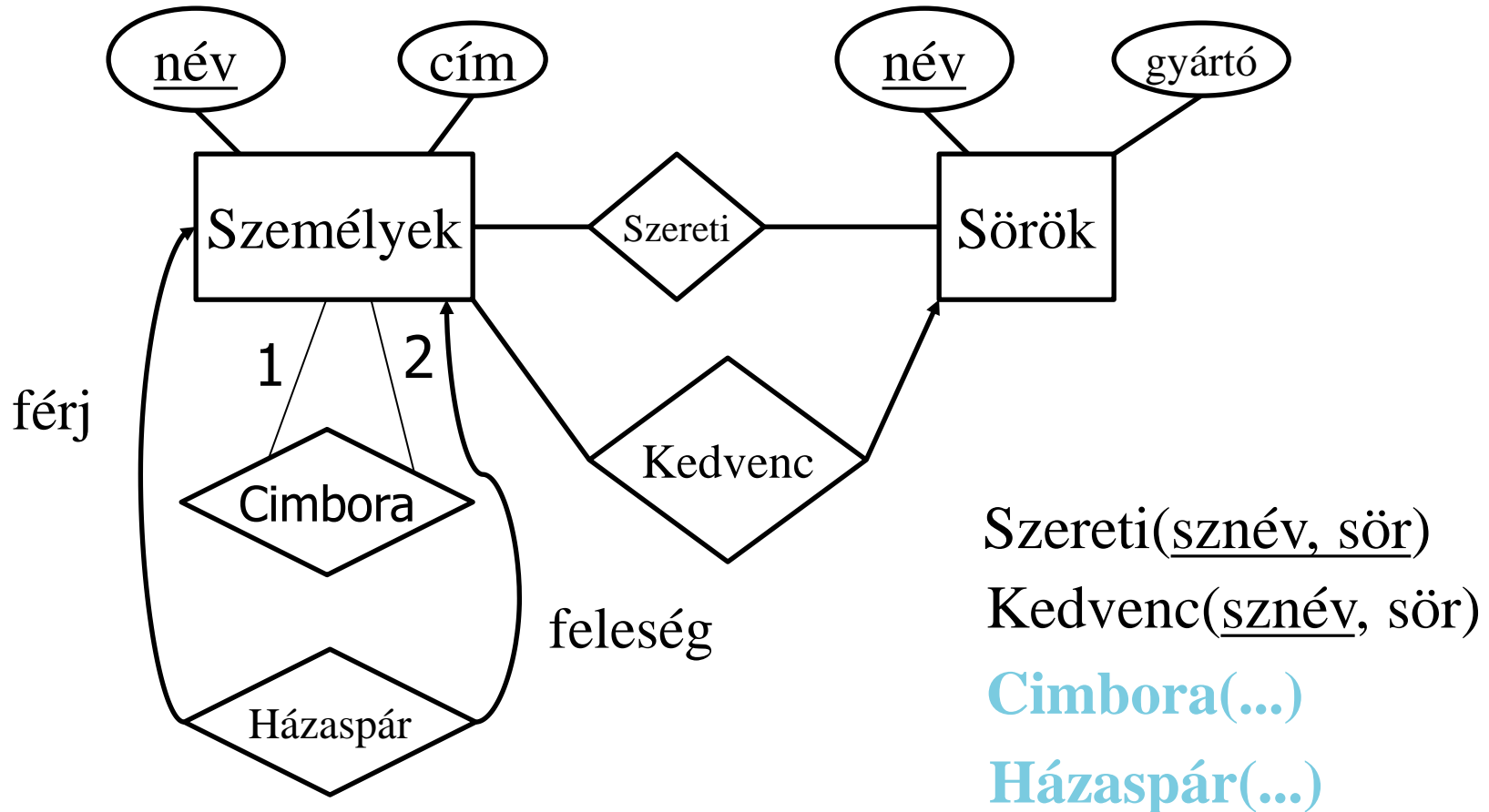


*Felt., hogy az alkalmazottaknak van főnökük.
Hogyan ábrázolhatjuk?*

Példa (Kapcsolat → Reláció)

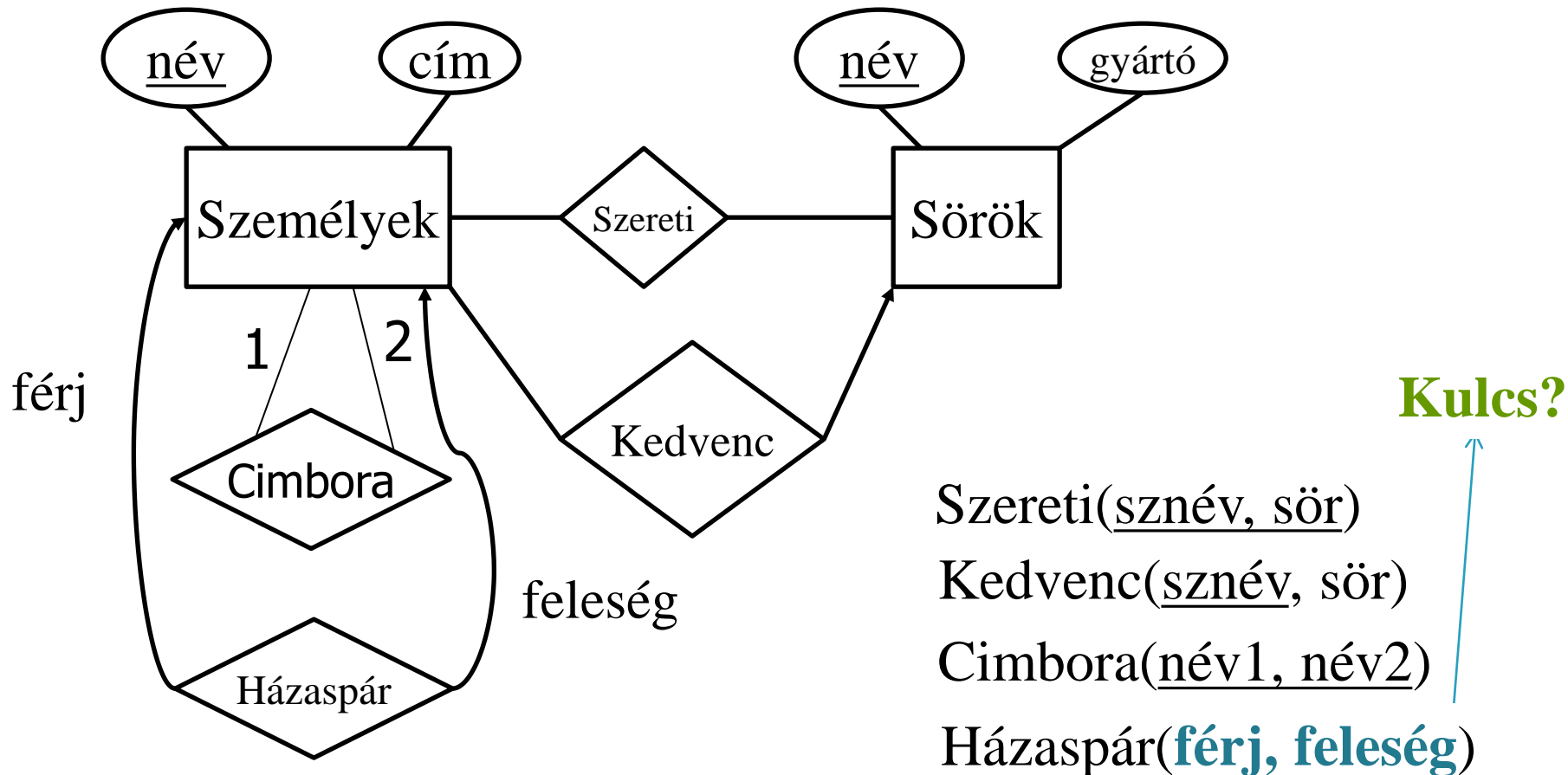


Példa (Kapcsolat → Reláció)



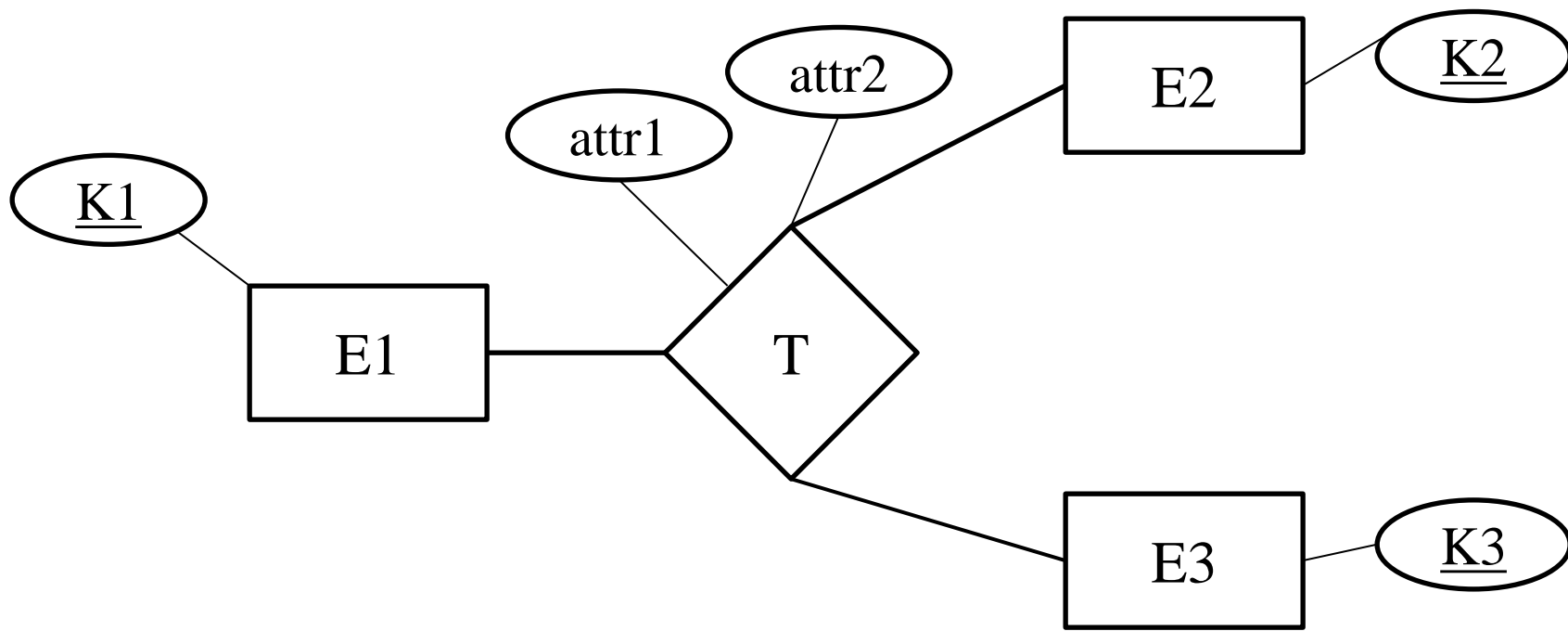
- A kapcsolatoknak megfelelő sémákban az oszlopokat célszerű átnevezni (pl. a szerepek alapján) + (név,név) séma nem szerepelhetne.

Példa (Kapcsolat → Reláció)

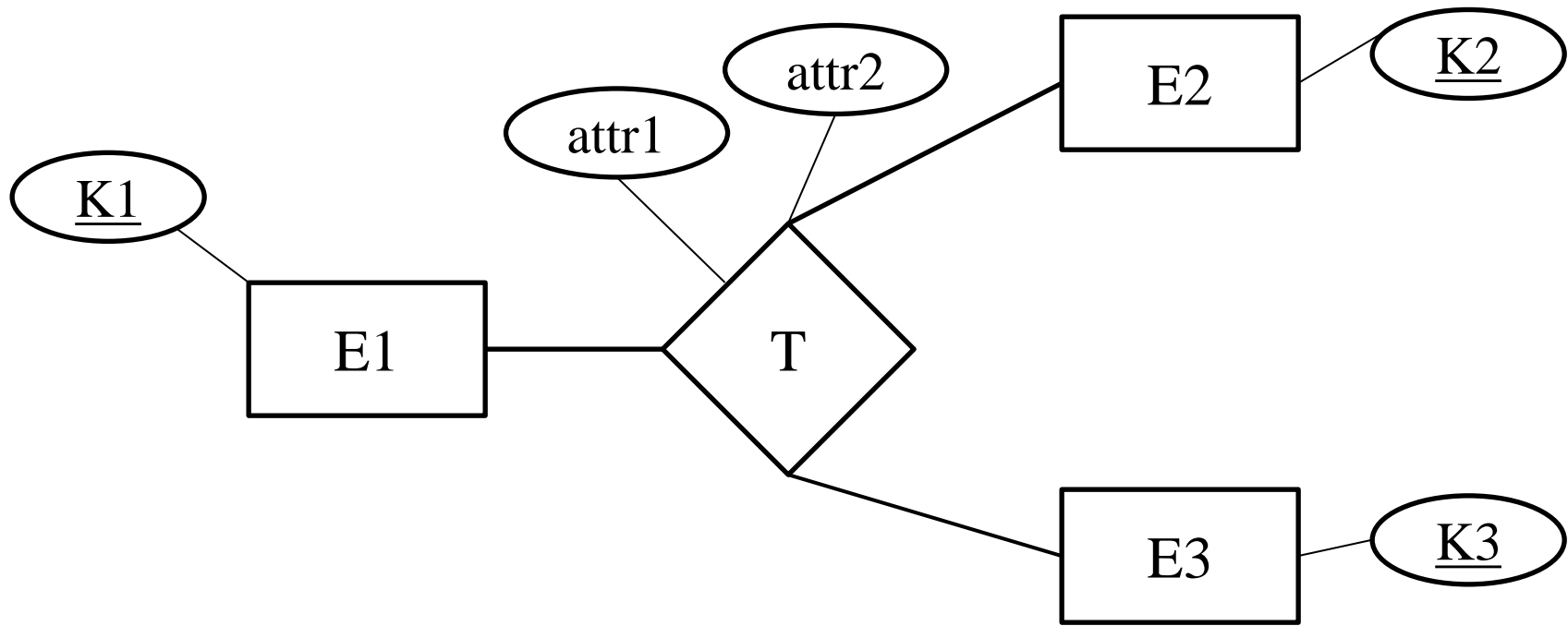


- A kapcsolatoknak megfelelő sémákban az oszlopokat célszerű átnevezni (pl. a szerepek alapján) + (név,név) séma nem szerepelhetne.

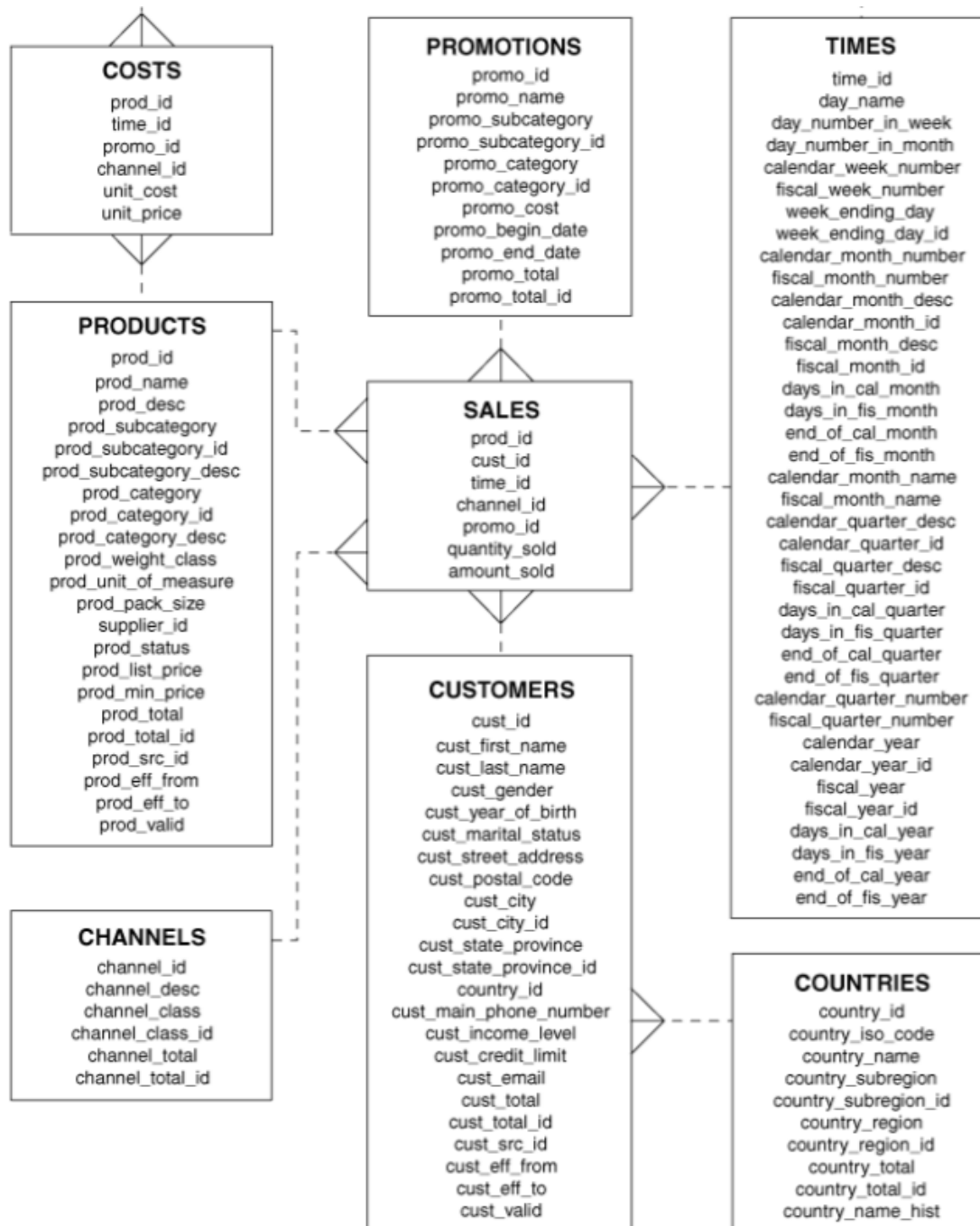
Többágú kapcsolat



Többágú kapcsolat



- $T(\underline{K1}, \underline{K2}, \underline{K3}, attr1, attr2)$
 - Nem vonódik össze.
 - Érdemes külön kulcsot definiálni, ha $n \geq 3$, ahol n a kapcsolat foka).



Többágú kapcsolatra példa

Oracle - Sales History séma

Osztályhierarchia átalakítása relációkká

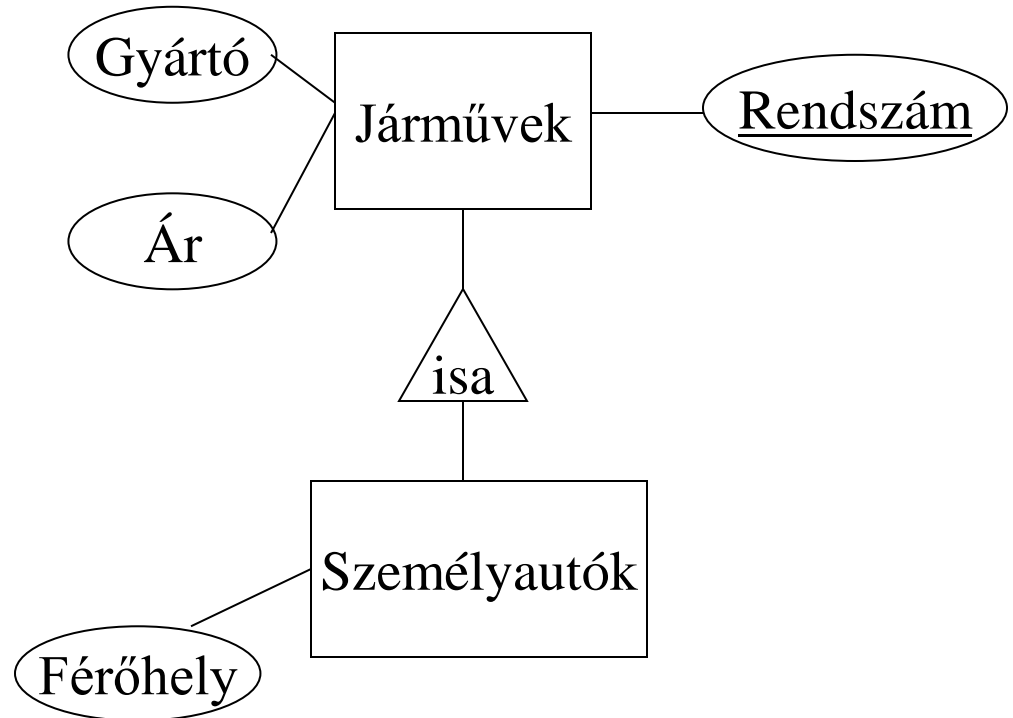
Három megközelítés:

- 1) *Az E/K szempontjainak követése*
- 2) *Objektumorientált megközelítés*
- 3) *Nullértékek használata*

Osztályhierarchia átalakítása relációkká

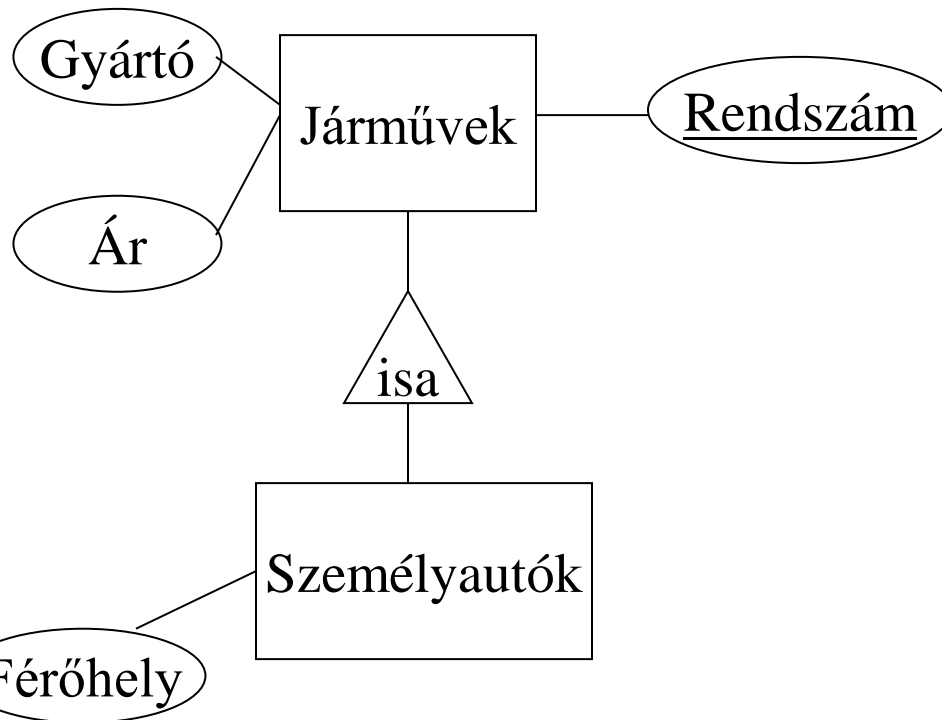
Három megközelítés:

- 1) *Az E/K szempontjainak követése*
- 2) *Objektumorientált megközelítés*
- 3) *Nullértékek használata*



Példa – E/K megközelítés

- Egy reláció minden alosztálynak a hierarchiában: kulcsattribútum(ok) + alosztály attribútumai.



Járművek

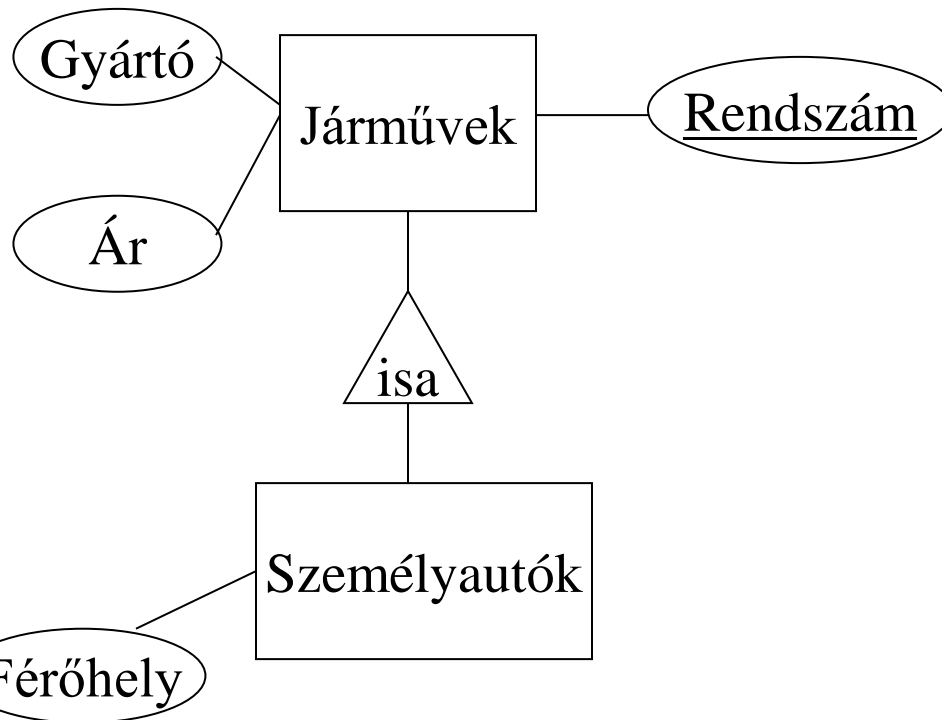
Rendszám	Gyártó	Ár
CJ-12ABC	Audi	10000
B-45VXZ	Man	45000

Személyautók

Rendszám	Férőhely
CJ-12ABC	5

Példa – E/K megközelítés

- Egy reláció minden alosztálynak a hierarchiában: kulcsattribútum(ok) + alosztály attribútumai.



Járművek

Rendszám	Gyártó	Ár
CJ-12ABC	Audi	10000
B-45VXZ	Man	45000

Személyautók

Rendszám	Férőhely
CJ-12ABC	5

P1. Találd meg az összes járművet (személyautókat is), amelyet az Audi gyárt.