

Egyed-kapcsolat modell

E/K diagramok

Gyenge egyedhalmazok

E/K diagramok átírása relációsémákká

Az E/K modell célja

- ◆ Az E/K modellel az adatbázissémát vázolhatjuk fel.
 - ◆ Bizonyos megszorításokat is megadhatunk, de műveleteket nem.
- ◆ A terveket *egyed-kapcsolat diagramnak* nevezzük.
- ◆ **Később**: az E/K diagramot relációs adatbázissémává alakítjuk..

Az E/K modell „kerete”

- ◆ A tervezés komoly része az AB építésnek.
- ◆ A “főnök” általában abban biztos, hogy szeretne egy adatbázist, de azt már nem feltétlen tudja, mi és hogyan legyen benne.
- ◆ A főbb részek felvázolásával hatékonyan lehet megtervezni egy működő adatbázist.

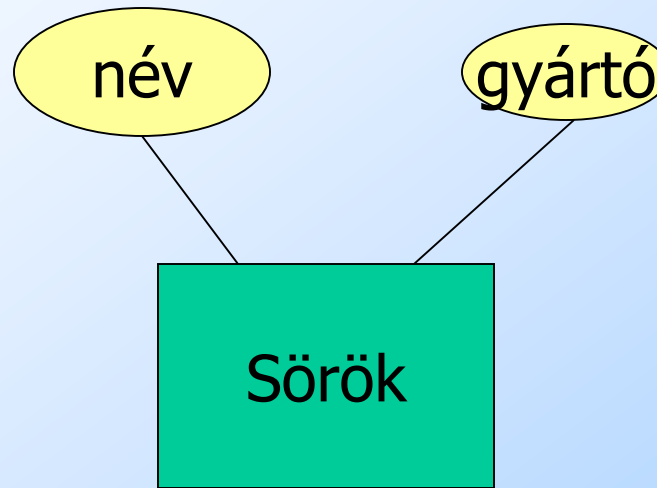
Egyedhalmazok

- ◆ *Egyed* = "valóságdarab" vagy objektum.
- ◆ *Egyedhalmaz* = hasonló egyedek kollekciója.
 - ◆ Az objektum-orientált nyelvek ***osztály foglamához*** hasonló.
- ◆ *Attribútum* = az egyedhalmaz egyedeinek egy tulajdonsága.
 - ◆ Az attribútumok atomi értékűek, például egészek (integer) vagy karakter sztringek, de nem rekordok, tömbök é.í.t.

E/K diagramok

- ◆ Az egyed-kapcsolat diagramban:
 - ◆ az egyedhalmaz = téglalap,
 - ◆ az attribútum = a megfelelő egyedhalmazt reprezentáló téglalappal összekötött ovális (krumpli).

Példa:

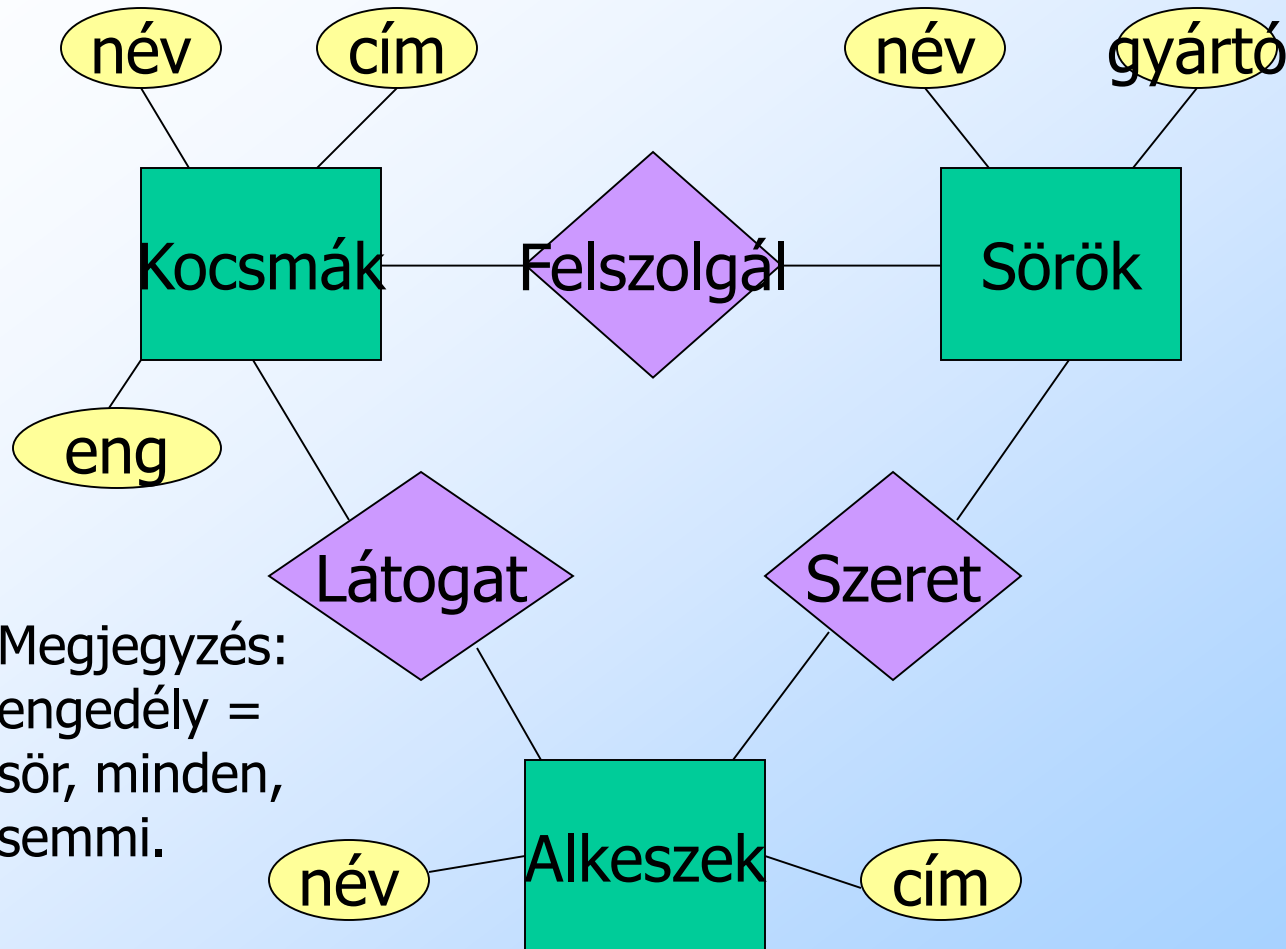


- ◆ A **Sörök** egyedhalmaznak két attribútuma van: **név** és **gyártó**.
- ◆ Minden **Sörök** entitás értékeket kap ezeken az attribútumokon, pl. (Bud, Anheuser-Busch).

Kapcsolatok

- ◆ Egy **kapcsolat** kettő vagy több egyedhalmazt köt össze.
- ◆ Rombusszal jelöljük, melyet a kapcsolatban részt vevő egyedhalmazok téglalapjaival kötünk össze.

Példa: kapcsolatok



Megjegyzés:
engedély =
sör, minden,
semmi.

Kocsmákban
sört árulnak.

Az alkeszek
szeretnek néhány
sört.

Emellett
kocsmákba
járnak.

Kapcsolat halmaz

- ◆ Az egyedhalmaz “értéke” a hozzá tartozó egyedek halmaza.
 - ◆ **Példa:** az adatbázisunkban az összes kocsmá halmaza.
- ◆ Egy kapcsolat “értéke” a *kapcsolat halmaz*, azaz sorok egy halmaza, melynek minden eleme egy a kapcsolatban résztvevő egyedhalmazból való.

Példa: kapcsolat halmaz

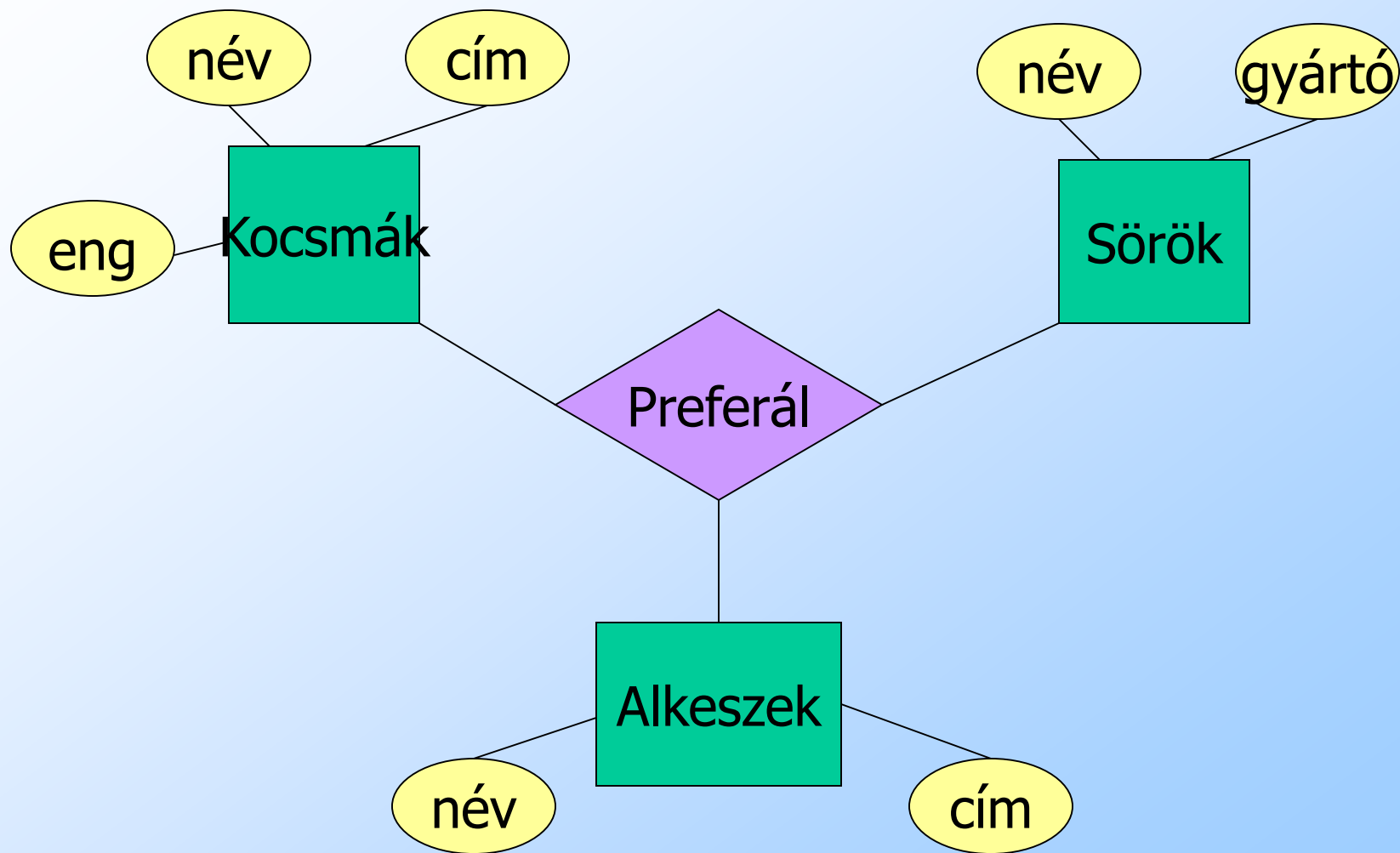
- ◆ A Felszolgál kapcsolathoz például az alábbi kapcsolat halmaz tartozhat:

Kocsma	Sör
Joe bárja	Bud
Joe bárja	Miller
Sue bárja	Bud
Sue bárja	Pete's Ale
Sue bárja	Bud Lite

Többirányú kapcsolatok

- ◆ Esetenként több, mint két egyedhalmazt összekötő kapcsolatra van szükség.
- ◆ Tegyük fel, hogy az egyes alkeszek bizonyos söröket csak bizonyos kocsmákban fogyasztanak.
 - ◆ A bináris kapcsolataink Szeret, Felszolgál és Látogat segítségével ezt nem tudjuk leírni.
 - ◆ De egy hármas kapcsolattal igen.

Példa: hármas kapcsolat



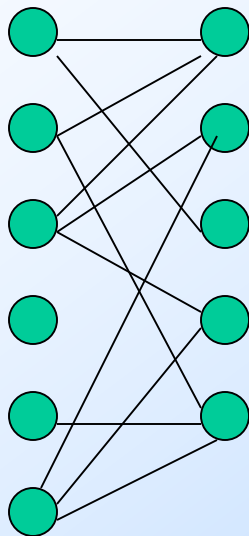
Egy tipikus kapcsolat halmaz

Kocsma	Alkesz	Sör
Joe bárja	Anna	Miller
Sue bárja	Anna	Bud
Sue bárja	Anna	Pete's Ale
Joe bárja	Pál	Bud
Joe bárja	Pál	Miller
Joe bárja	Karcsi	Miller
Sue bárja	Karcsi	Bud Lite

Sok-sok kapcsolat

- ◆ Elsősorban a **bináris** kapcsolatok esetén értelmezzük, mint például a **Kocsmák** és **Sörök** közötti **Felhasználó** kapcsolat.
- ◆ A **sok-sok kapcsolatokban**, mind a két résztvevő egyedhalmaz egyedei több másik egyedhez kapcsolódhatnak.
 - ◆ A példában: egy kocsmában több sör árulható; egy sör is több kocsmában árulható.

Képen

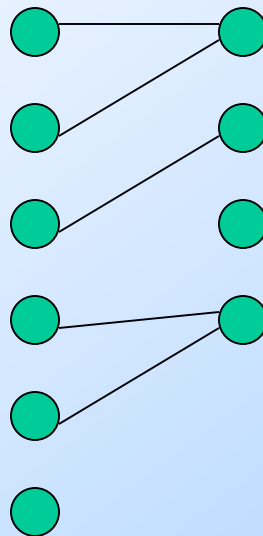


sok-sok

Sok-egy kapcsolat

- ◆ Ezen kívül léteznek még *sok-egy* kapcsolatok.
- ◆ Minden egyede az első egyedhalmaznak legfeljebb egy egyedhez kapcsolódhat a másik halmazból.
- ◆ De a második halmaz entitásai nulla, egy vagy több entitáshoz is kapcsolódhatnak az első halmazból.

Képen



sok-egy

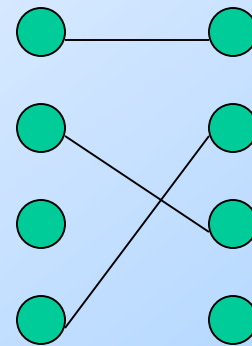
Példa: sok-egy kapcsolat

- ◆ Az **Alkeszekből** a **Kocsmákba** menő **Kedvenc (Preferál)** kapcsolat sok-egy típusú.
- ◆ Egy alkesznek legfeljebb egy kedvenc söre lehet.
- ◆ De egy sör több vagy akár nulla alkesznek is a kedvence lehet.

Egy-egy kapcsolatok

- ◆ Egy *egy-egy kapcsolat* esetén minden egyes entitás legfeljebb egyetlen másik entitáshoz kapcsolódhat.
- ◆ Példa: a **Best-seller** kapcsolat a **Gyártók** és **Sörök** egyedhalmazok között.
 - ◆ Egy sört nem gyárthat több cég és egy gyártónak sem lehet több bestseller söre.

Képen

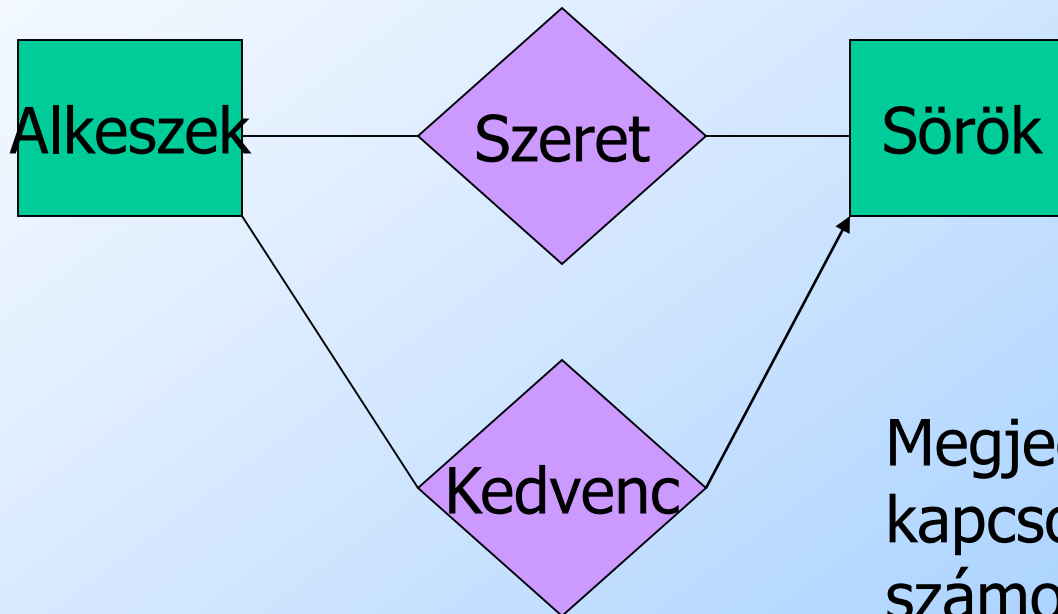


egy-egy

A „számosság” jelzése

- ◆ A sok-egy kapcsolat „egy oldalát” egy nyíl jelzi.
- ◆ Az egy-egy kapcsolatok mindkét végén nyíl van.
- ◆ A **lekerekített nyíl** jelentése: „pontosan egy”, azaz minden első halmazbeli entitásnak pontosan egy párja van a második egyedhalmazból.

Példa: sok-egy kapcsolat

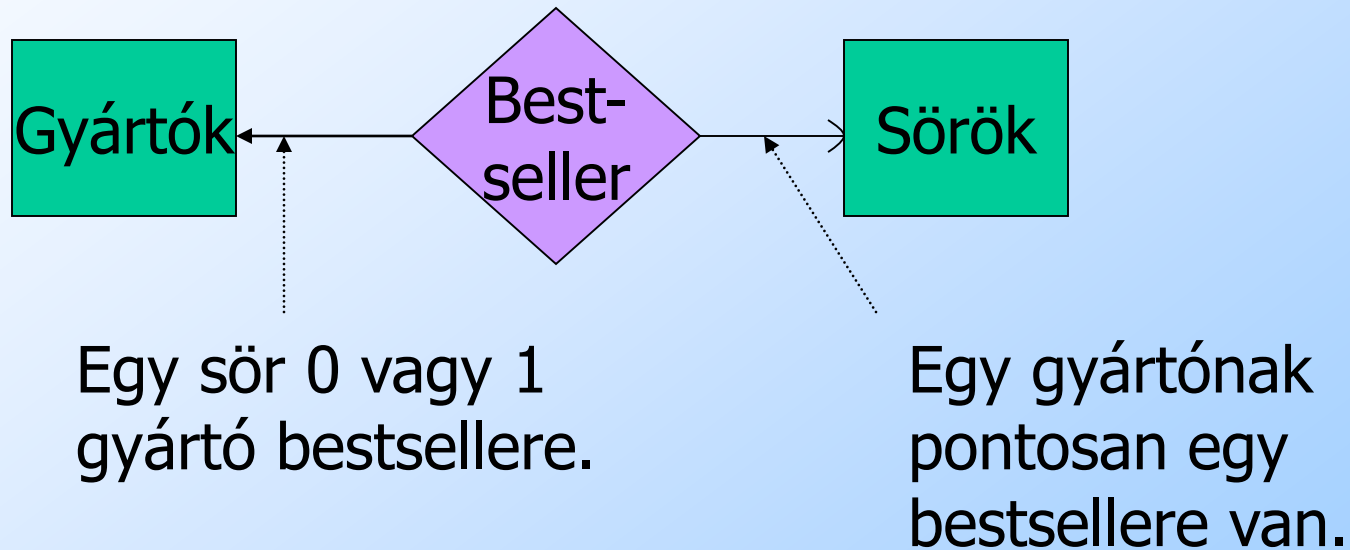


Megjegyzés: a két kapcsolat különböző számosságú.

Példa: egy-egy kapcsolat

- ◆ Vegyük a Gyártók és Sörök közötti Bestseller kapcsolatot.
- ◆ Néhány sör valószínűleg egyetlen gyártónak sem lesz bestsellere, tehát a Gyártók egyedhalmaz felőli nyíl nem lehet kerek.
- ◆ De egy gyártónak kell lennie bestsellerének.

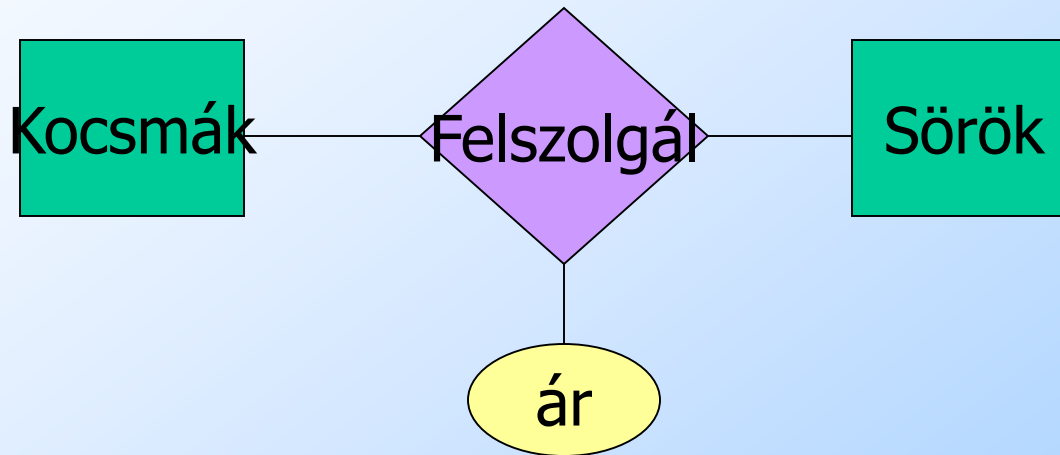
Az E/K diagramon



A kapcsolatok attribútumai

- ◆ Néha hasznos ha a kapcsolathoz is illeszthetünk attribútumot.
- ◆ Erre úgy tekinthetünk, mint a kapcsolat halmazban lévő sorok egy tulajdonságára.

Példa: kapcsolat attribútuma

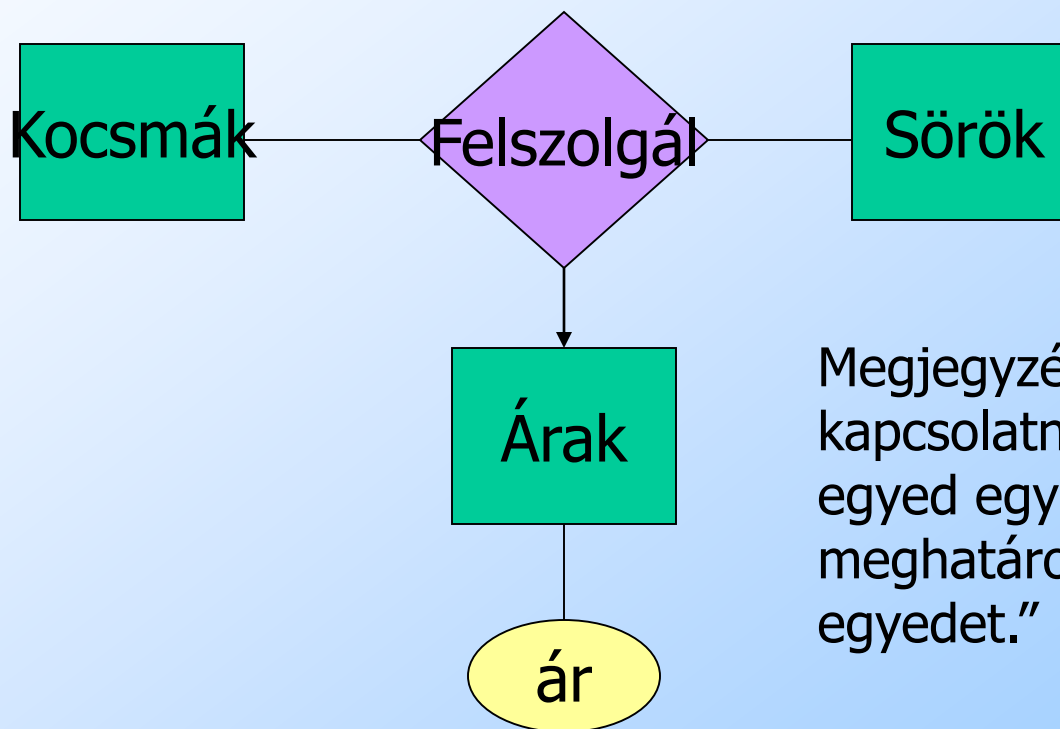


Az ár a kocsmának és a sörnek együttes függvénye.

Egy másik lehetőség kapcsolatok attribútumának ábrázolására

- ◆ „Vegyünk fel” egy egyedhalmazt, ami a kérdéses attribútum értékeit reprezentálja.
- ◆ Ezt az új egyedhalmazt „vonjuk be” a szóban forgó kapcsolatba.

Példa: kapcsolat attribútumának átírása



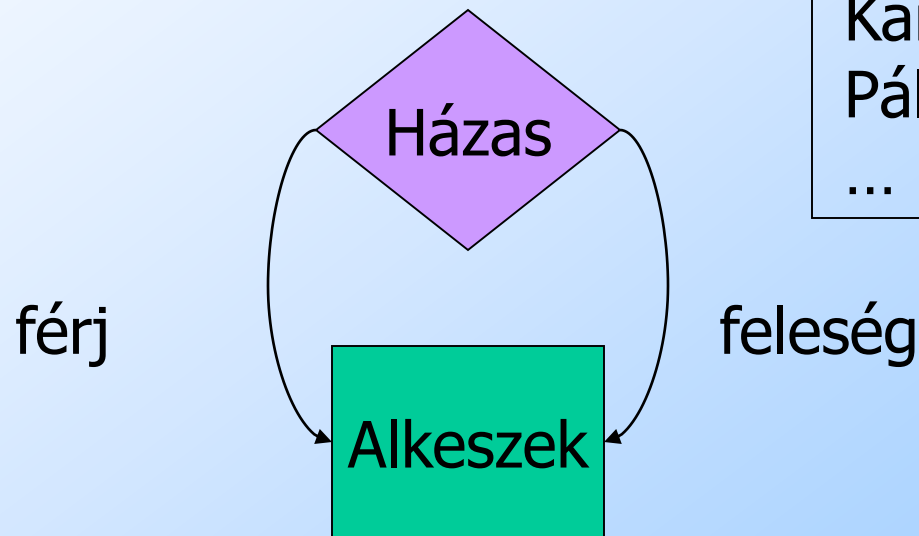
Megjegyzés: nyíl egy többirányú kapcsolatnál = „a többi egyed együttese egyértelműen meghatározza a szóban forgó egyedet.”

Szerepek

- ◆ Néha egy egyedhalmaz többször is megjelenik egy kapcsolatban.
- ◆ Ilyenkor a megfelelő éleket címkézzük és a címkéket *szerepeknek* nevezzük.

Példa: szerepek

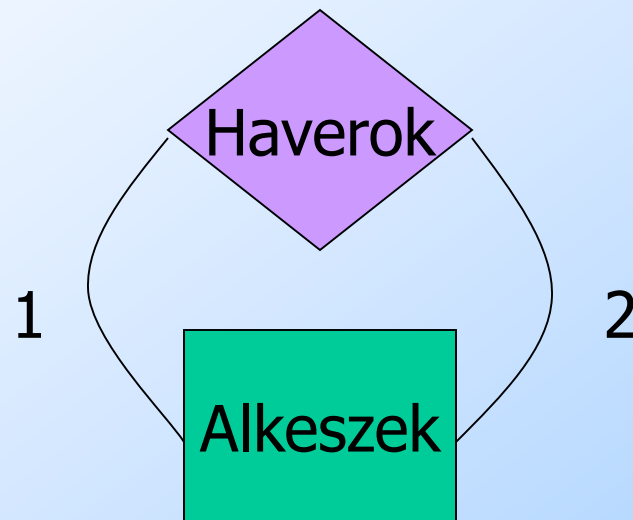
Kapcsolat halmaz



Férj	Feleség
Karcsi Pál	Anna Éva
...	...

Példa: szerepek

Kapcsolat halmaz



Haver1	Haver2
Pál	Anna
Rezső	Éva
Éva	Karcsi
Richárd	Jenő
...	...

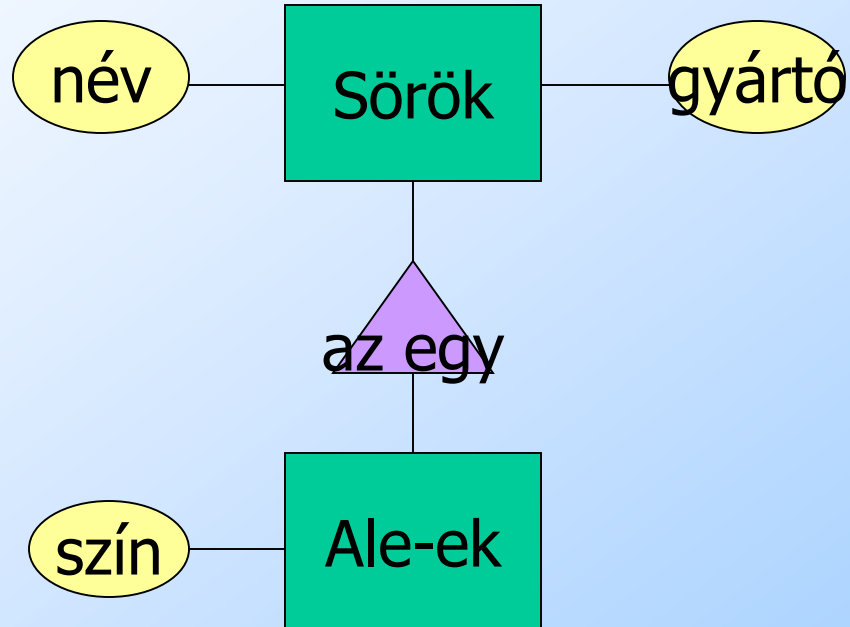
Alosztályok

- ◆ *Alosztály* = speciális eset = kevesebb egyed = több tulajdonság (kapcsolat).
- ◆ Példa: Az Ale-ek mind sörök is.
 - ◆ Nem minden sör ale (*angol barna*), de minden ale sör.
 - ◆ Tegyük fel, hogy a sörök már meglévő *tulajdonságai* (attribútumok és kapcsolatok) mellé az ale-ek esetében még egy a *szín* attribútumot is felveszünk.

Alosztályok E/K diagramokon

- ◆ Feltesszük, hogy az osztályok rendszere fát alkot.
 - ◆ Azaz, nincs többszörös öröklés.
- ◆ Az osztály kapcsolatokat az-egy háromszögek jelölik.
 - ◆ Az őssztályra mutat a háromszög felső csúcsa.

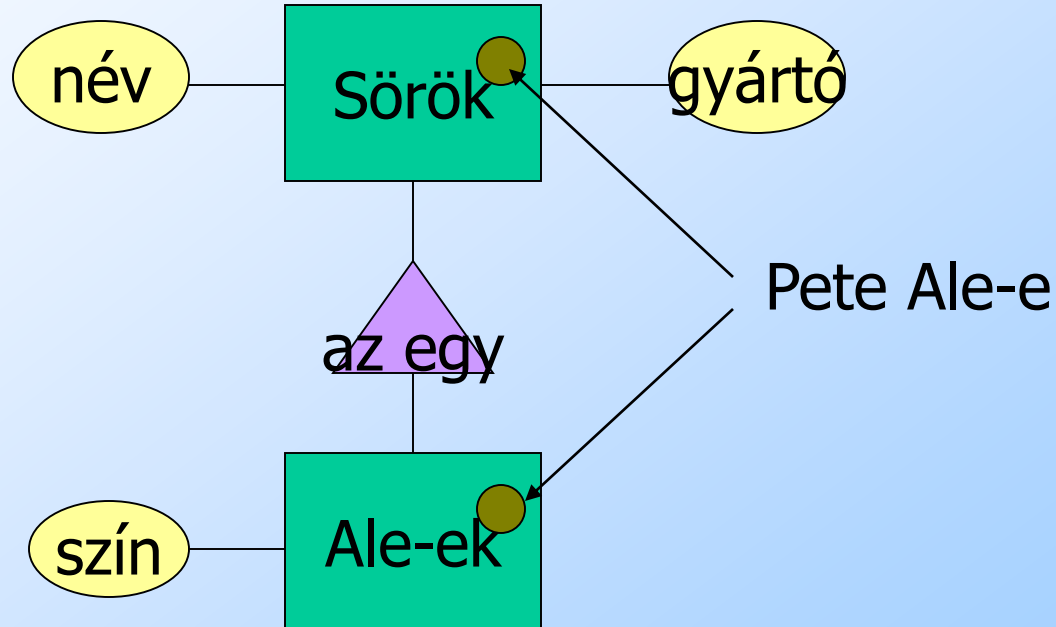
Példa: részosztályok



E/K vs. objektumorientált (OO) részosztályok

- ◆ Az OO paradigmában minden objektum pontosan egy osztálynak lehet eleme.
 - ◆ A részosztályok az ősosztályoktól örökölnék.
- ◆ Ezzel ellentétben az E/K entitásoknak minden részosztályban vannak *reprezentánsai*, amihez hozzátartoznak.
 - ◆ Szabály: ha az e entitás szerepel egy részosztályban, akkor e szerepel az ősosztály(ok)ban is.

Példa: entitások előfordulásai



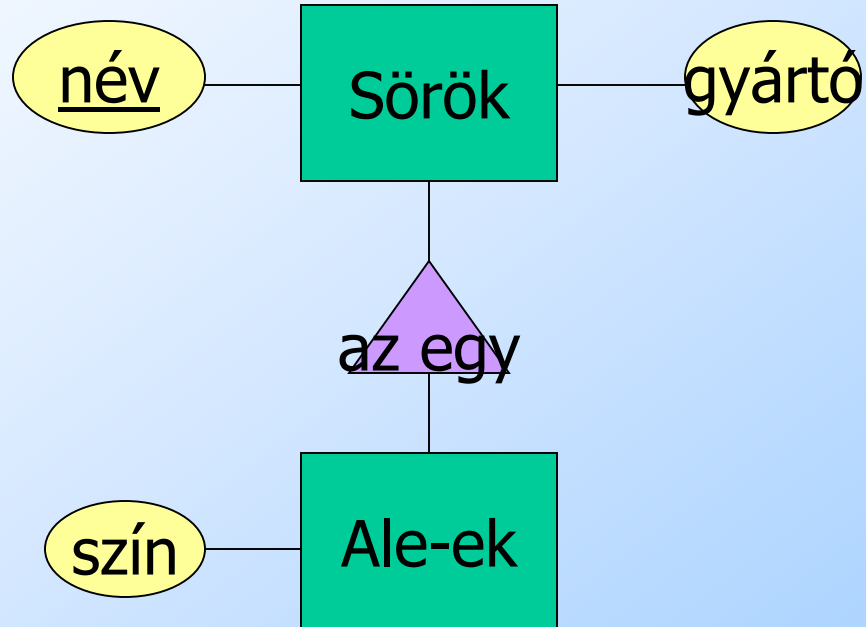
Kulcsok

- ◆ A *kulcs* az attribútumoknak egy olyan halmaza, amelyekre nem létezhet két olyan entitás, amelyek a kulcsattribútumok mindegyikén azonos értéket vennének fel.
 - ◆ Ugyanakkor a kulcs néhány attribútumán megegyezhetnek, de az összesen nem.
- ◆ Minden entitáshalmazhoz meg kell adnunk egy kulcsot.

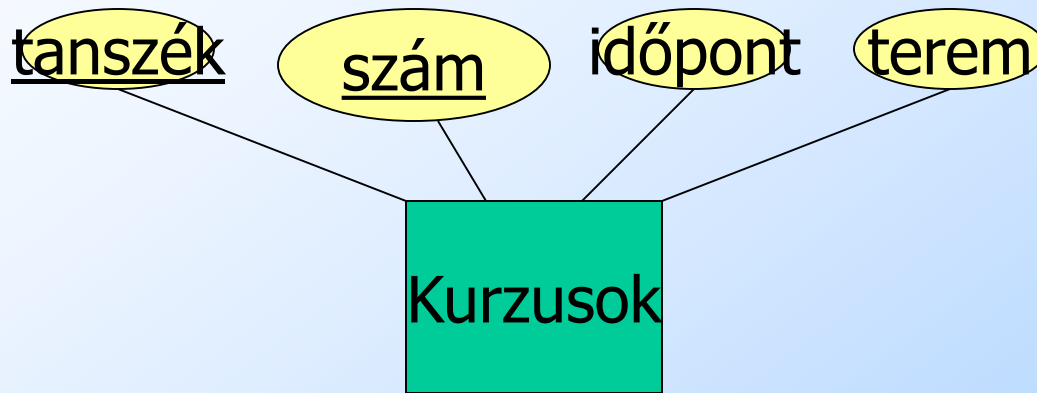
Kulcsok az E/K diagramokon

- ◆ A kulcsattribútomo(ka)t aláhúzással jelöljük.
- ◆ Egy öröklődési hierarchiában csak a gyökér entitáshalmaznak lehet kulcsa, ez lesz a hierarchiában szereplő többi alosztálynak is a kulcsa.

Példa: a **név** kulcs a Sörökben



Példa: több attribútumú kulcs



- Az **időpont** és **terem** attribútumok együtt szintén kulcsot alkotnak, a modellben azonban csak egy kulcsot adhatunk meg.

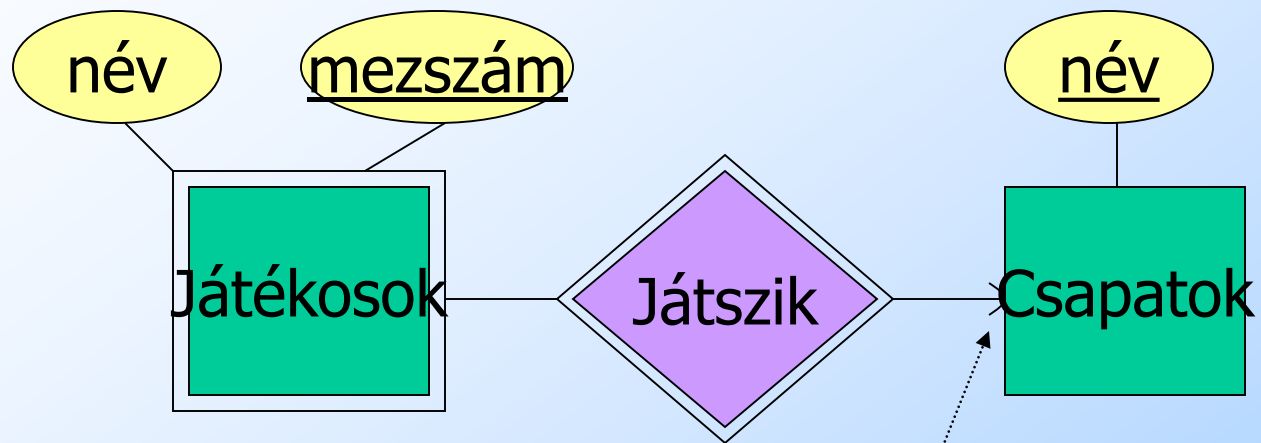
Gyenge egyedhalmazok

- ◆ Esetenként egy-egy egyedhalmaz egyedeit csak “külső segítséggel” lehet egyértelműen azonosítani.
- ◆ Egy E egyedhalmazt *gyengének* nevezünk, ha ahhoz, hogy E elemeit azonosítsuk, egy vagy több, E -ből induló sok-egy kapcsolatot követve a kapcsolódó egyedek kulcsértékeire is szükségünk van.

Példa: gyenge egyedhalmazok

- ◆ a **név** majdnem kulcs a focisták esetén, ritkán azonban előfordulhat, hogy két játékosnak ugyanaz a neve.
- ◆ a **mezzszám** nyilván nem kulcs.
- ◆ Ám a **mezzszám** a csapat **nevével** kombinálva a **Játszik** kapcsolaton keresztül már egyedi minden játékos esetén.

Az E/K diagramon



Megjegyzés: itt minden játékoshoz kell, hogy tartozzon csapat.

- A gyenge egyedhalmazt dupla téglalap jelzi.
- A *támogató* sok-egy kapcsolatot dupla rombuszsal jelöljük.

Gyenge egyedhalmaz szabályok

- ◆ Egy gyenge egyedhalmaznak egy vagy több sok-egy kapcsolata lehet más (támogató) egyedhalmazokhoz.
 - ◆ Nem az összes sok-egy kapcsolatnak kell támogatónak lennie.
 - ◆ De a támogató kapcsolatoknak kerek nyílban kell végződniük az egy oldalon (azaz minden entitásnak a gyenge egyedhalmazból pontosan egy egyedhez kell kapcsolódnia a támogató egyedhalmazból).

Gyenge egyedhalmaz szabályok – (2)

- ◆ A gyenge egyedhalmaz kulcsa saját aláhúzott és a támogató egyedhalmaz(ok) aláhúzott attribútumaiból áll.
 - ◆ Például a (játékos) **mezszáma** és a (csapat) **neve** kulcs lesz a **Játékosok** egyedhalmazban.

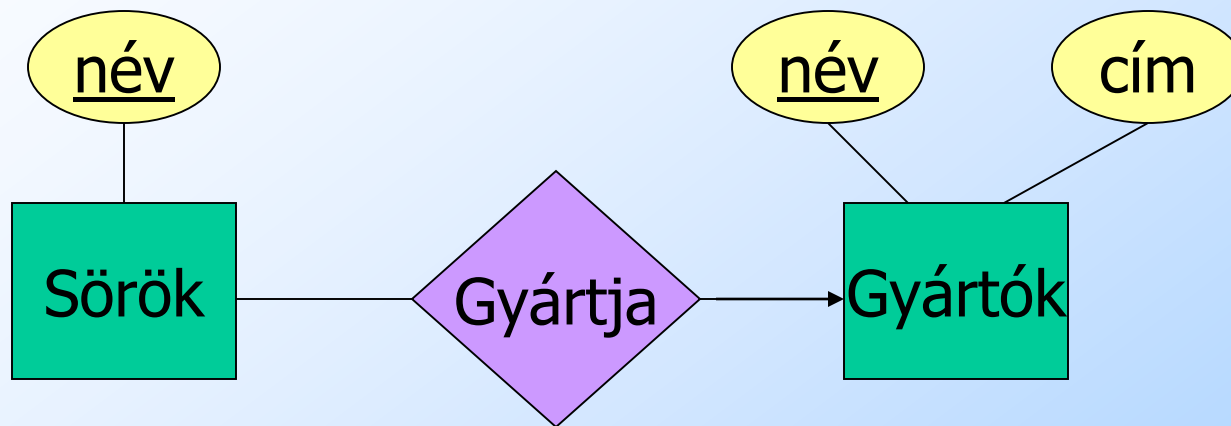
Tervezési technikák

1. Redundancia elkerülése.
2. A gyenge egyedhalmazok óvatos használata.
3. Ne használjunk egyedhalmazt, ha egy attribútum éppúgy megfelelne a célnak.

Redundancia elkerülése

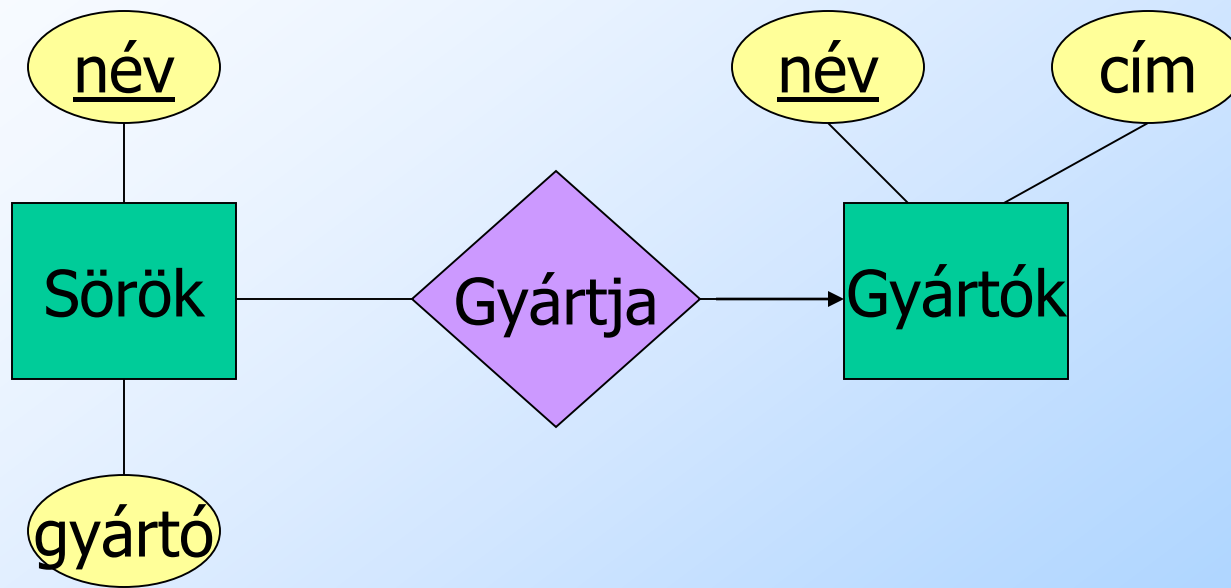
- ◆ *Redundancia* = ugyanazt a dolgot többféle módon is feltüntetjük.
- ◆ Ez egyrészt helypazarlás, másrészt növeli az inkonzisztencia veszélyét.
 - ◆ Ugyanazon tény két leírása akkor válik inkonzisztenssé, ha az egyik értéket megváltoztatjuk, míg a másiktól megfeledezünk.
 - ◆ Emlékezzünk vissza a módosítási anomáliára.

Példa: helyes



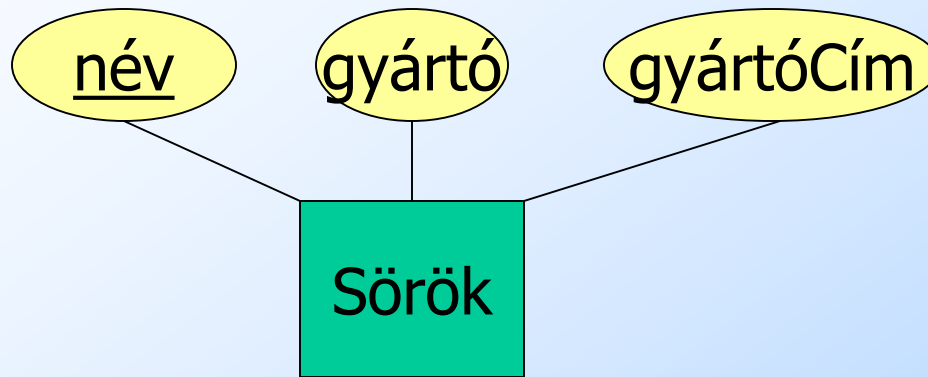
Ennél a diagramnál minden gyártónál pontosan egyszer szerepel a címe.

Példa: helytelen



Ennél a diagramnál viszont egy-egy sör gyártója kétszer is szerepel: attribútumként és a kapcsolódó Gyártók egyedhalmazon keresztül.

Példa: helytelen



Ez a diagram minden sör esetén felsorolja a gyártó címét. Ha pedig ideiglenesen nincs megadva az adott gyártóhoz sör, akkor elveszítjük ezt a címet.

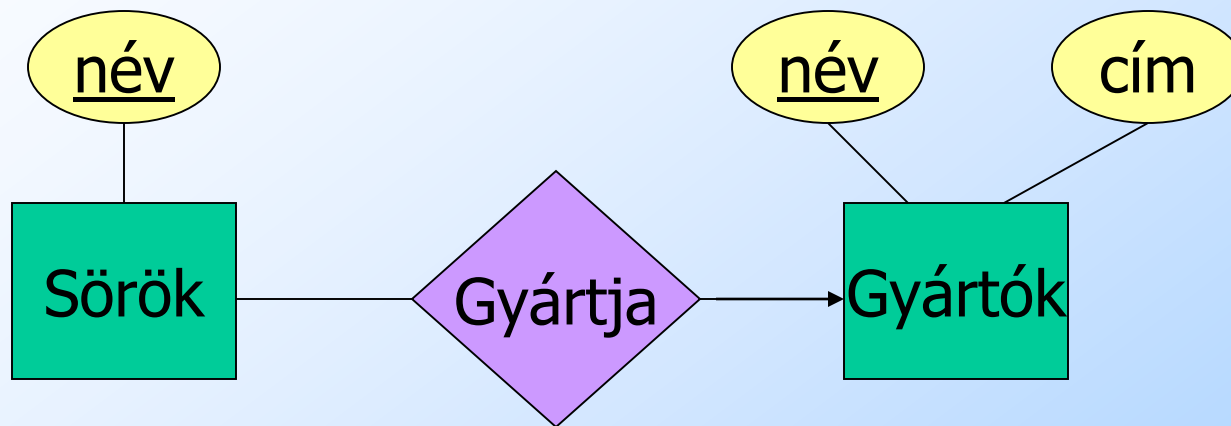
Egyedhalmazok vs. attribútumok

- ◆ Egy egyedhalmaznak legalább egy feltételnek eleget kell tennie az alábbiak közül:
 - ◆ Többnek kell lennie, mint egy egyszerű név, azaz legalább egy nem kulcs attribútumának lennie kell.

Vagy..

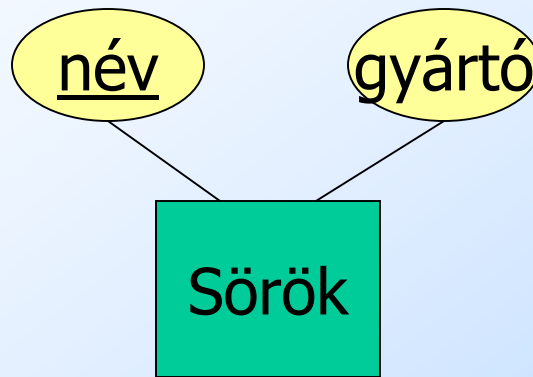
- ◆ a „sok” végén szerepel egy sok-egy kapcsolatnak.

Példa: helyes



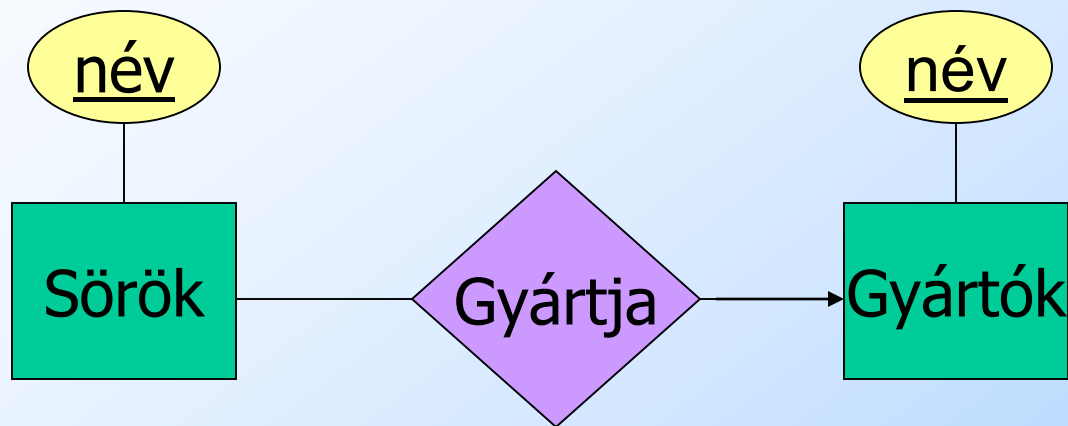
- A **Gyártóknak** valóban egyedhalmaznak kell lennie a nem kulcs **cím** attribútum miatt.
- A **Sörökre** ugyanez igaz, hiszen a **Gyártja** kapcsolat „sok” végén szerepel.

Példa: helyes



Itt nem szükséges külön egyedhalmazt definiálni a gyártók számára, hiszen a nevükön kívül mást nem tárolunk.

Példa: rossz



Mivel a gyártóknál csak a nevet tároljuk, ezen kívül a sok-egy kapcsolat „egy” oldalán szerepel, ezért fölösleges külön egyedhalmazt definiálni a számára.

Ne használjunk gyenge egyedhalmazokat fölöslegesen

- ◆ Kezdő adatbázis tervezőknek gyakran kétségeik vannak abban tekintetben, hogy mit is lehet kulcsként definiálni, ami megállna önmaga jogán kulcsként.
 - ◆ Minden egyedhalmazt gyengének definiálnak, amit az összes vele összekötött egyedhalmaz támogat.
- ◆ A „nagybetűs életben” gyakran külön azonosítókat készítenek az egyedhalmazokhoz.
 - ◆ Pl. TAJ, rendszám tábla stb.

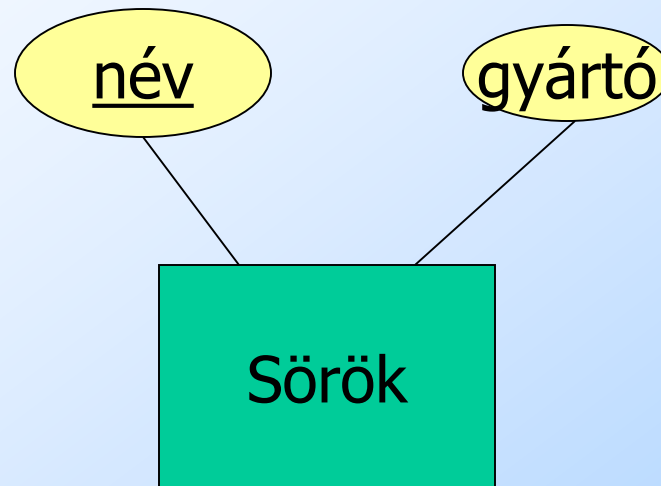
Mikor van szükség gyenge egyedhalmazokra?

- ◆ A legtöbb esetben nem létezik olyan globális fórum, amely kioszthatná az egyedi azonosítókat.
- ◆ **Példa:** nem valószínű, hogy el lehetne érni, hogy a világ összes focijátékosát külön mezszámmal regisztrálják.

E/K diagramok átírása relációsémába

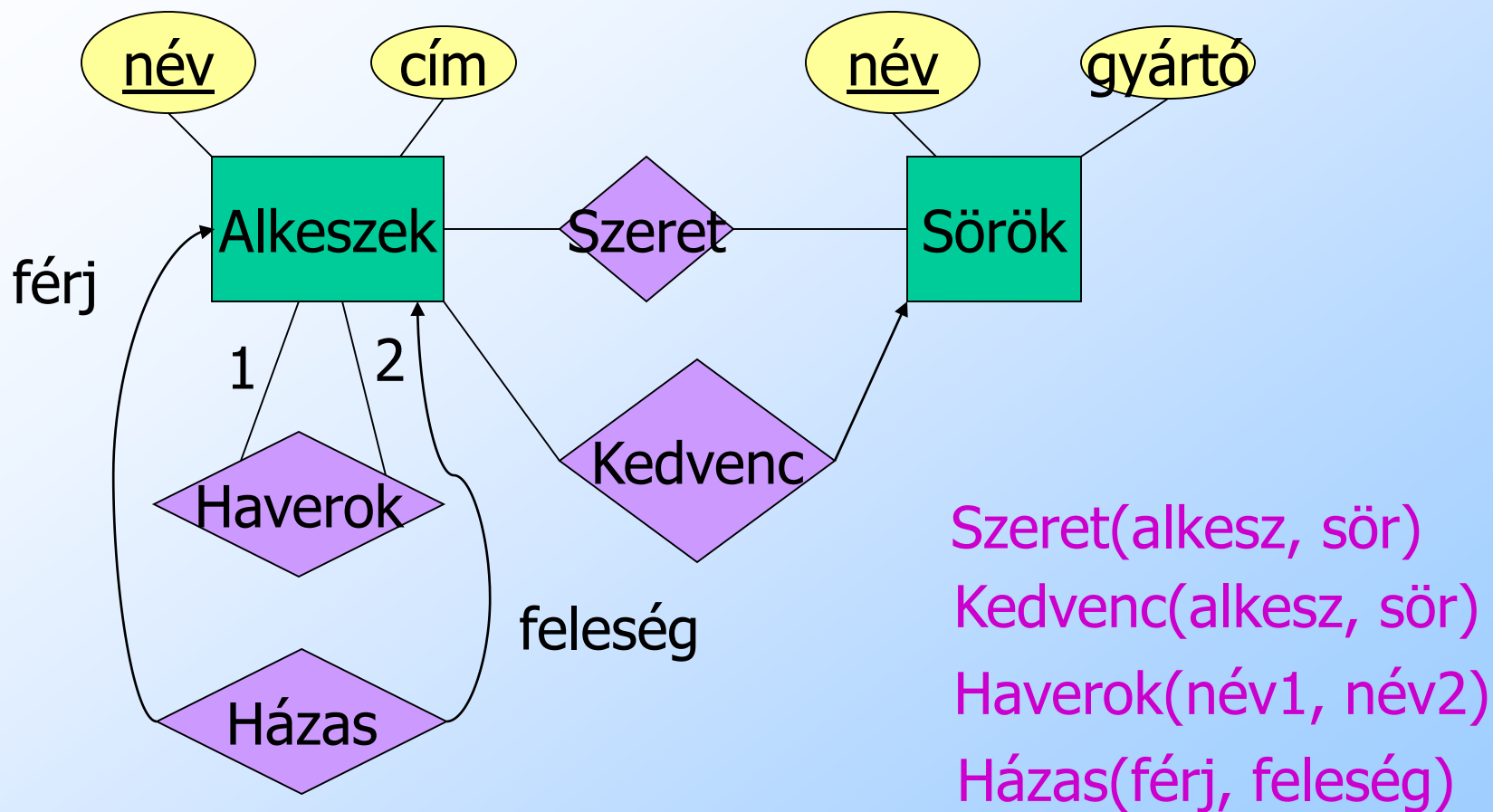
- ◆ Egyedhalmaz -> reláció.
 - ◆ Attribútumok -> attribútumok.
- ◆ Kapcsolat -> relációk, melyeknek az attribútumai csak:
 - ◆ az összekapcsolt egyedhalmazok kulcs-attribútumait,
 - ◆ és a kapcsolat attribútumait tartalmazzák.

Egyedhalmaz -> Reláció



Reláció: **Sörök(név, gyártó)**

Kapcsolat -> Reláció



Relációk összevonása

- ◆ Egy relációba összevonhatók:
 1. Az E egyedhalmazból kapott reláció,
 2. valamint azon sok-egy kapcsolatok relációi, melyeknél az E a „sok” oldalon szerepel.
- ◆ **Példa:** $\text{Alkeszek}(\text{név}, \text{cím})$ és $\text{Kedvenc}(\text{alkesz}, \text{sör})$ összevonható az $\text{Alkeszek1}(\text{név}, \text{cím}, \text{kedvencSör})$ relációvá.

A sok-sok kapcsolat nem összevonható

- ◆ Az Alkeszek és Szeret relációk összevonása hiba lenne, redundanciához vezet.

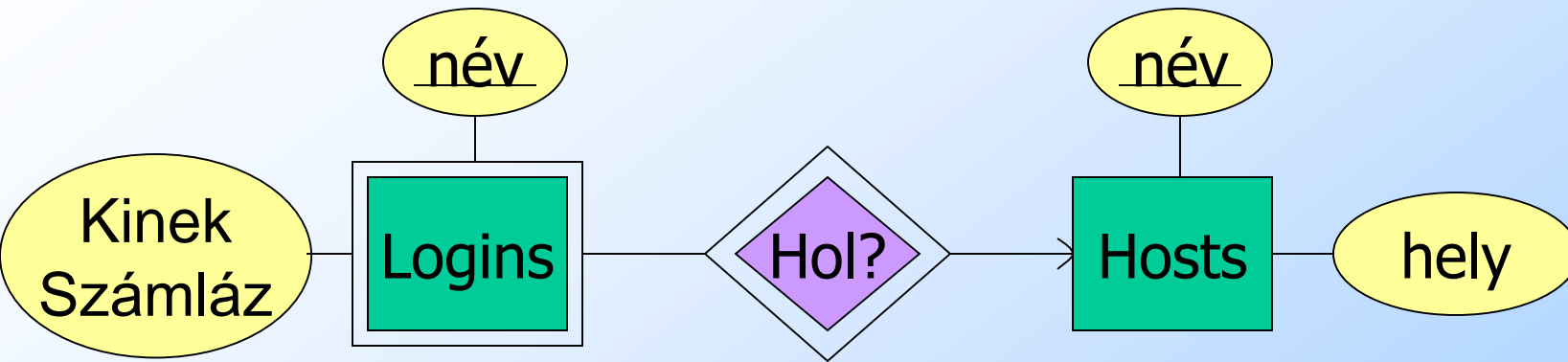
név	cím	Sör
Zsuzsa	123 Maple	Bud
Zsuzsa	123 Maple	Miller

Redundancia

Gyenge egyedhalmazok átírása

- ◆ Egy gyenge egyedhalmazokból kapott relációnak a teljes kulcsot tartalmaznia kell (a más egyedhalmazokhoz tartozó kulcs-attribútumokat is), valamint a saját, további attribútumokat.
- ◆ A támogató kapcsolatot nem írjuk át, redundanciához vezetne.

Példa: gyenge egyedhalmaz -> reláció



Hosts(hostNév, hely) (GazdaGép, Kiszolgáló)

Logins(loginNév, hostNév, billTo)

~~Hol(loginNév, hostNév, hostNév2)~~

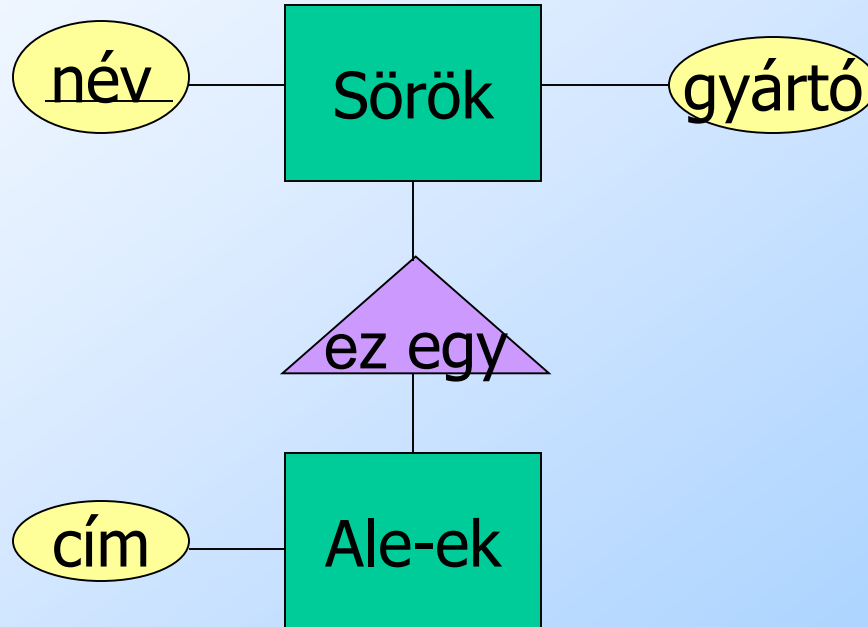
Meg kell egyezniük.

A Hol a Logins
részévé válik.

Entitás Alosztályok: három megközelítés

1. ***Objektumorientált*** : minden alosztályhoz egy reláció tartozik az összes releváns attribútummal.
2. ***Null értékek használata*** : az öröklődési hierarchiát egyetlen reláció reprezentálja, az entitások a rájuk nem vonatkozó attribútumoknál null értéket kapnak.
3. ***E/K style***: minden alosztályhoz külön reláció tartozik:
 - ◆ a kulcs-attribútummal (kulcs-attribútumokkal).
 - ◆ a részosztály további attribútumaival.

Példa: részosztály -> reláció



Objektumorientált

név	gyártó
Bud	Anheuser-Busch

Sörök

név	gyártó	szín
Summerbrew	Pete's	barna

Ale-ek

A „Pete által gyártott ale-ek színeit” visszaadó
lekérdezések esetén hasznos

E/K Style

név	gyártó
Bud	Anheuser-Busch
Summerbrew	Pete's

Sörök

név	szín
Summerbrew	barna

Ale-k

A Pete által gyártott söröket visszaadó
lekérdezések esetén hasznos.

Null értékek használata

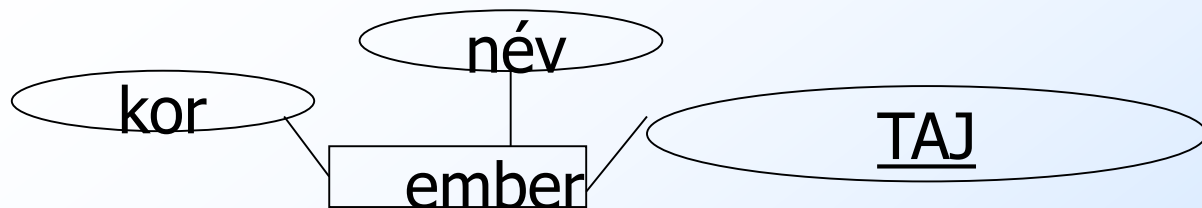
név	gyártó	szín
Bud Summerbrew	Anheuser-Busch Pete's	NULL barna

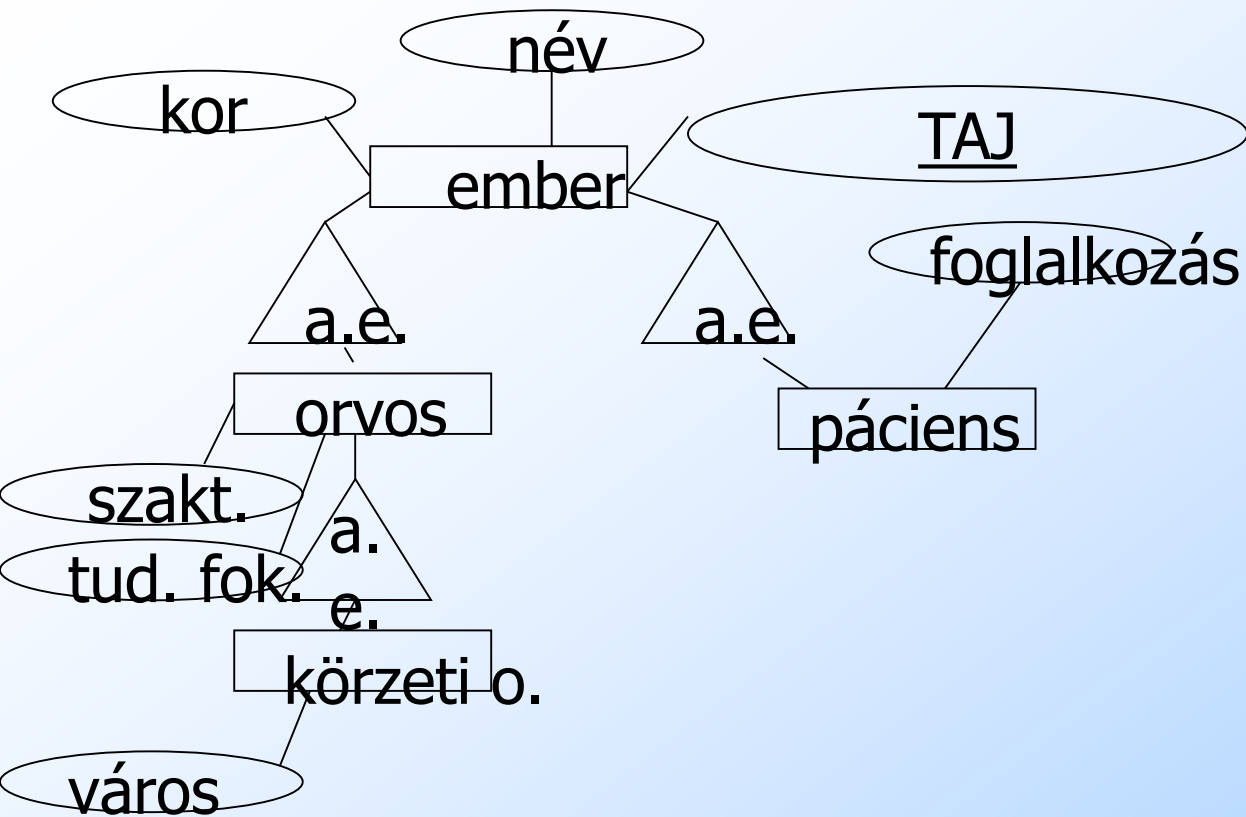
Sörök

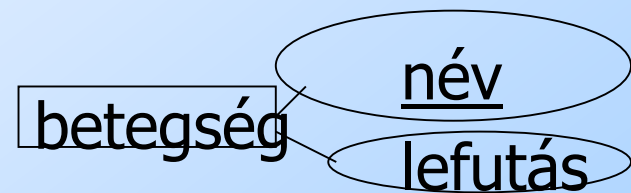
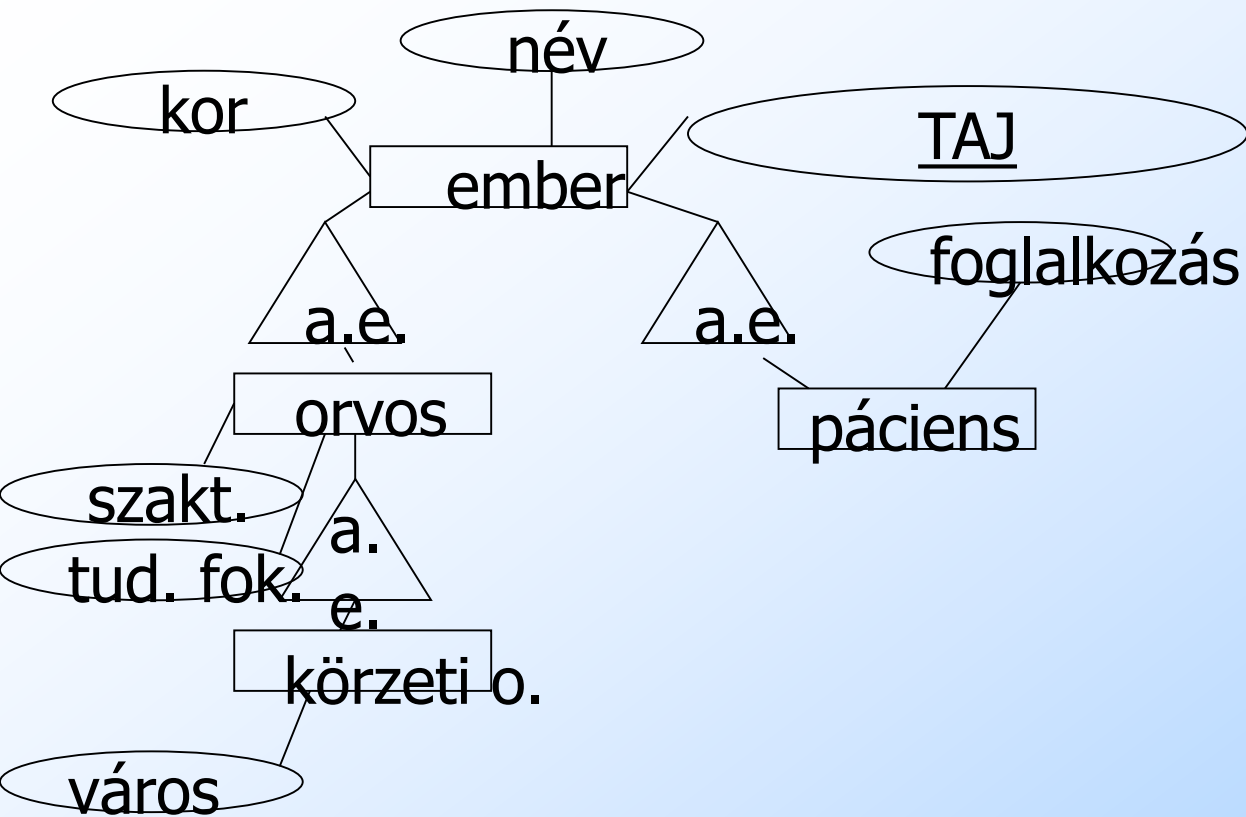
Helytakarékos, ha csak nem túl sok a NULL érték. Összekapcsolásokat is megspórolhatunk.

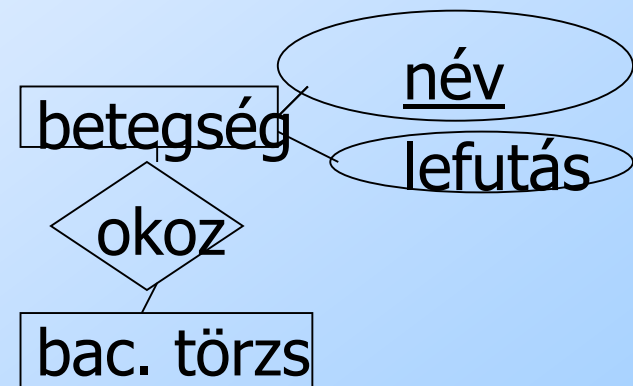
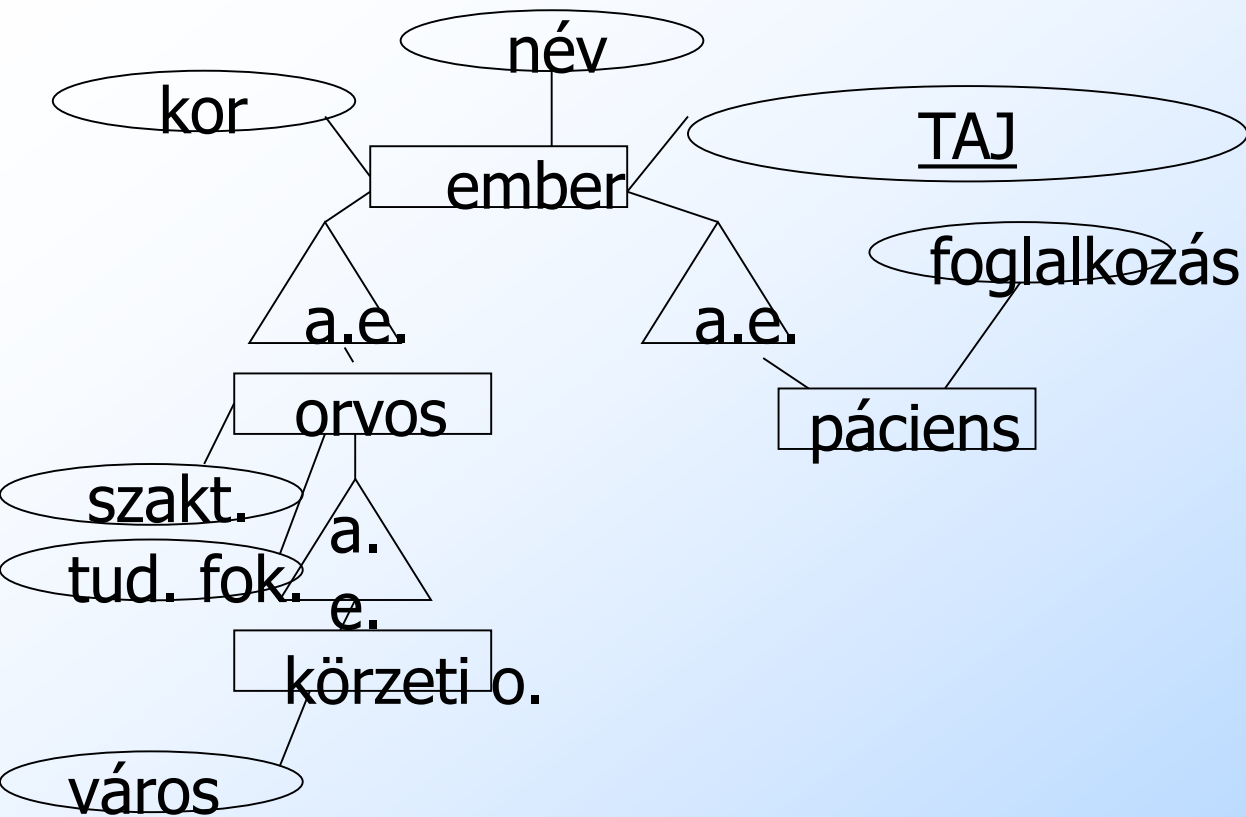
Feladat

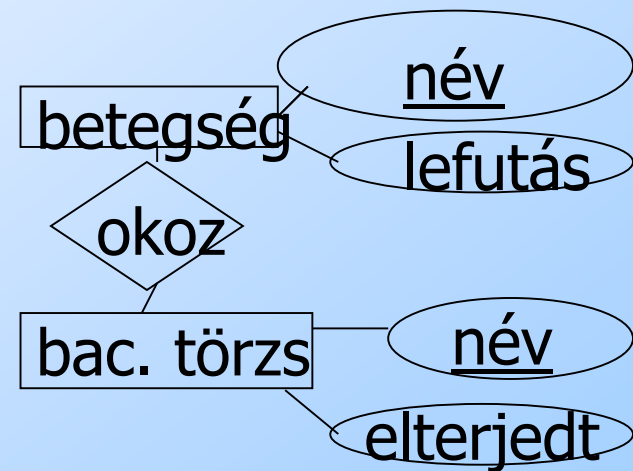
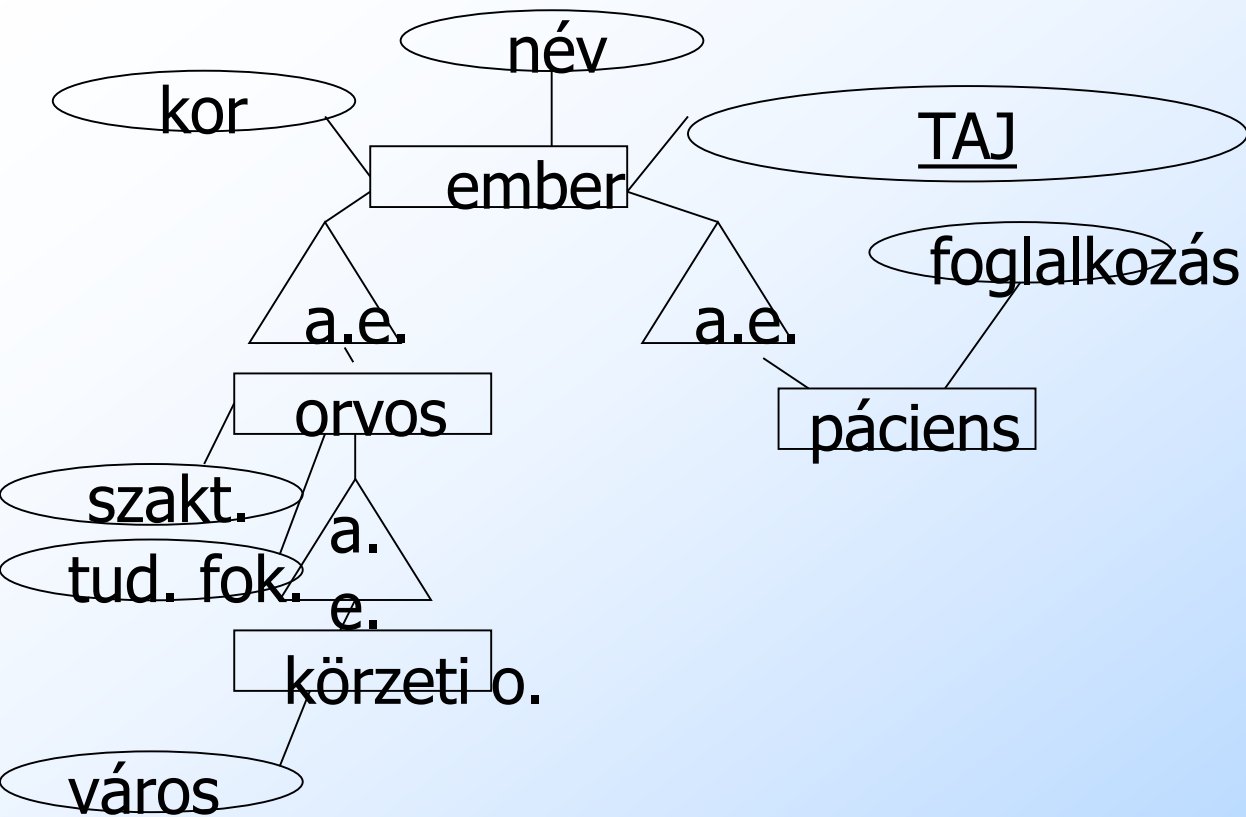
- ◆ Orvosi adatbázist készítünk. Minden embernél számontartjuk a nevét, korát, TAJ számát. Ezen utóbbi alapján egyértelműen azonosítani lehet bárkit. Az orvosoknál tároljuk ezeken kívül még a tudományos fokozatukat és a szakterületüket, a körzeti orvosoknál még annak a városnak a nevét is, ahol rendelnek, a pácienseknél pedig a foglalkozásukat. A betegségeknél számontartjuk a nevüket és azt, hogy átlagosan mennyi ideig tart a gyógyulási folyamat. Minden betegséget valamilyen bacilustörzs okoz. Ezek a nevük alapján egyértelműek, emellett tároljuk az elterjedtségük arányát. A törzsekhez bacilusok tartoznak, ám a név alapján még nem tudhatjuk pontosan melyik bacilusról van szó, mert több törzshöz is tartozhat ugyanolyan nevű bacilus. A különféle betegségekben szenvedő pácienseknek lehet, hogy egyszerre több orvos írja majd fel a különféle árú gyógyszereket. A gyógyszereknél tároljuk a nevüket, és hogy mely bacilusok esetében hatásosak. Minden páciensnek tartoznia kell egy körzeti orvoshoz, s valaki csak úgy lehet körzeti orvos, ha legalább húsz páciense van.

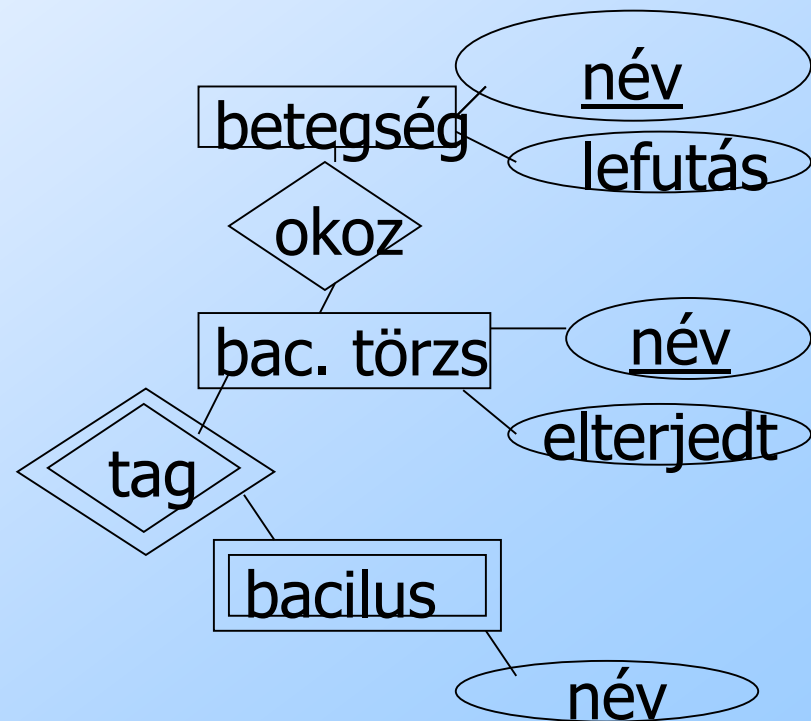
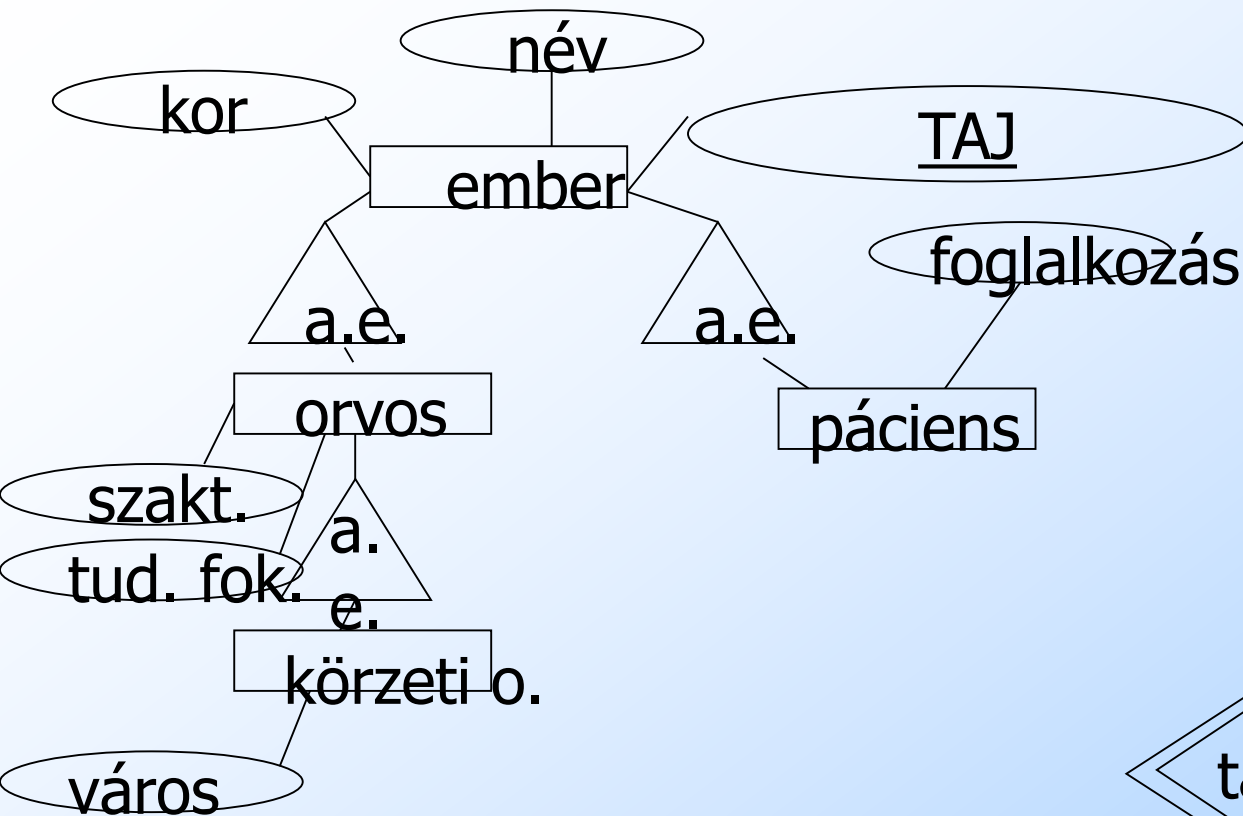


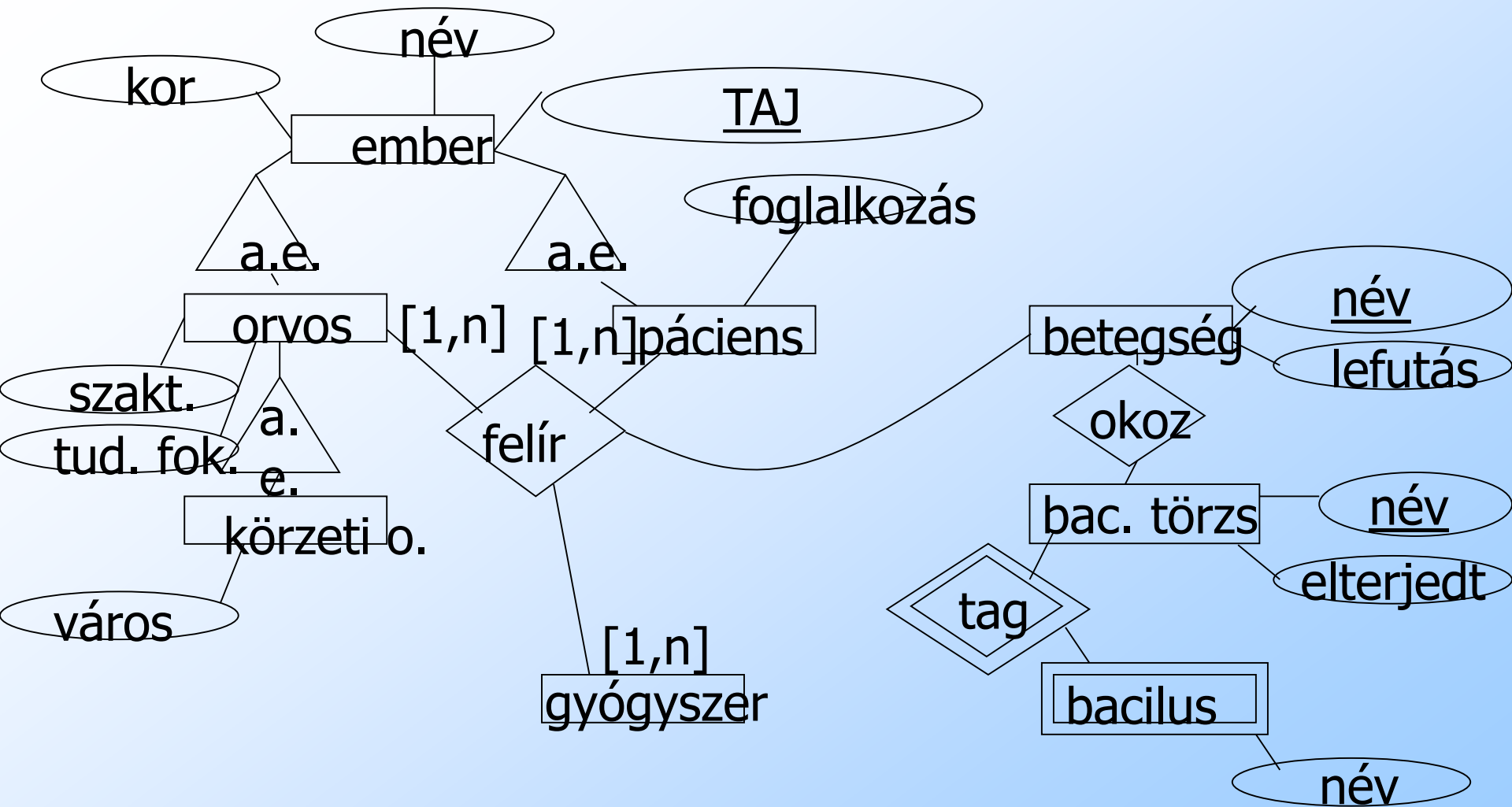


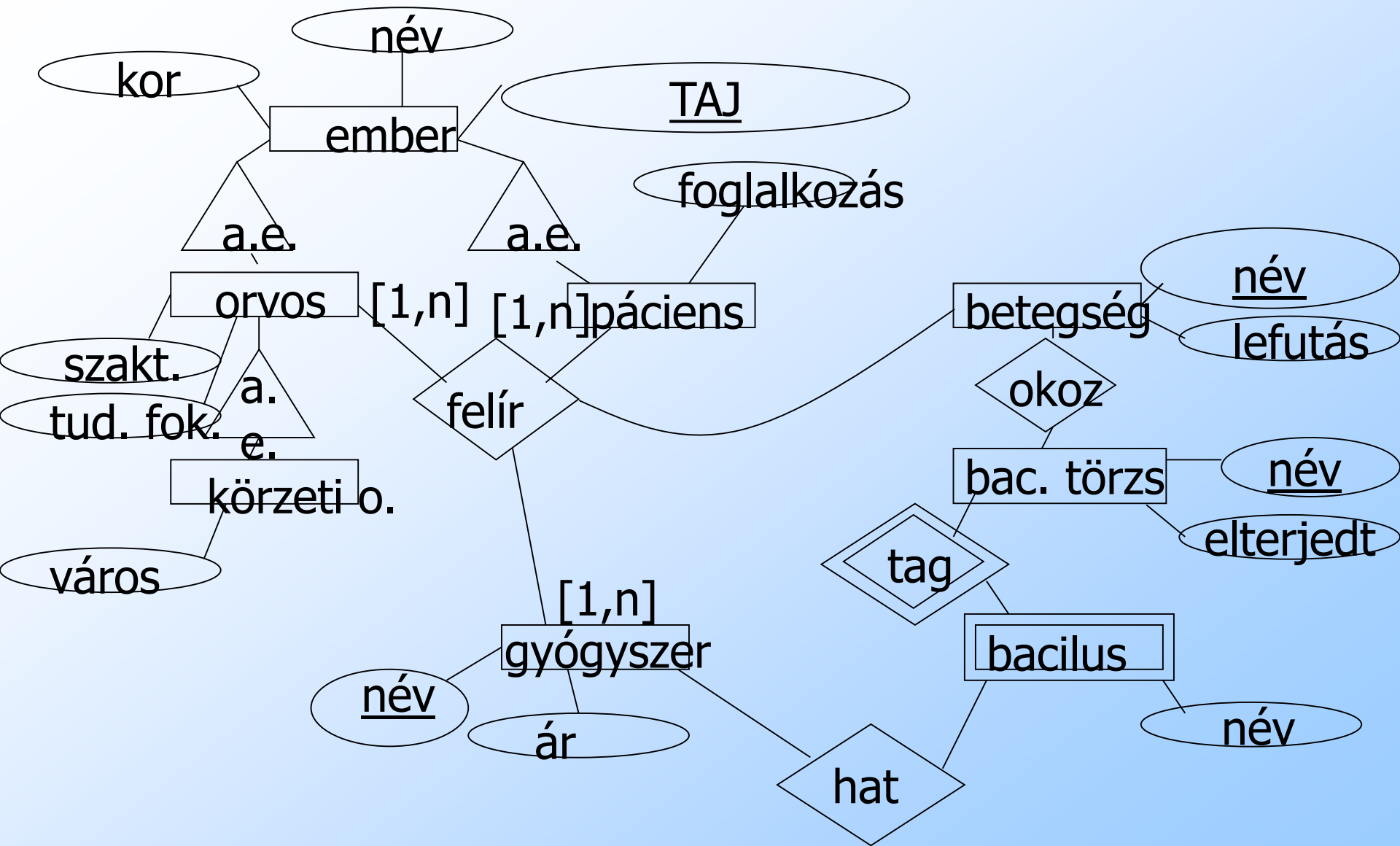


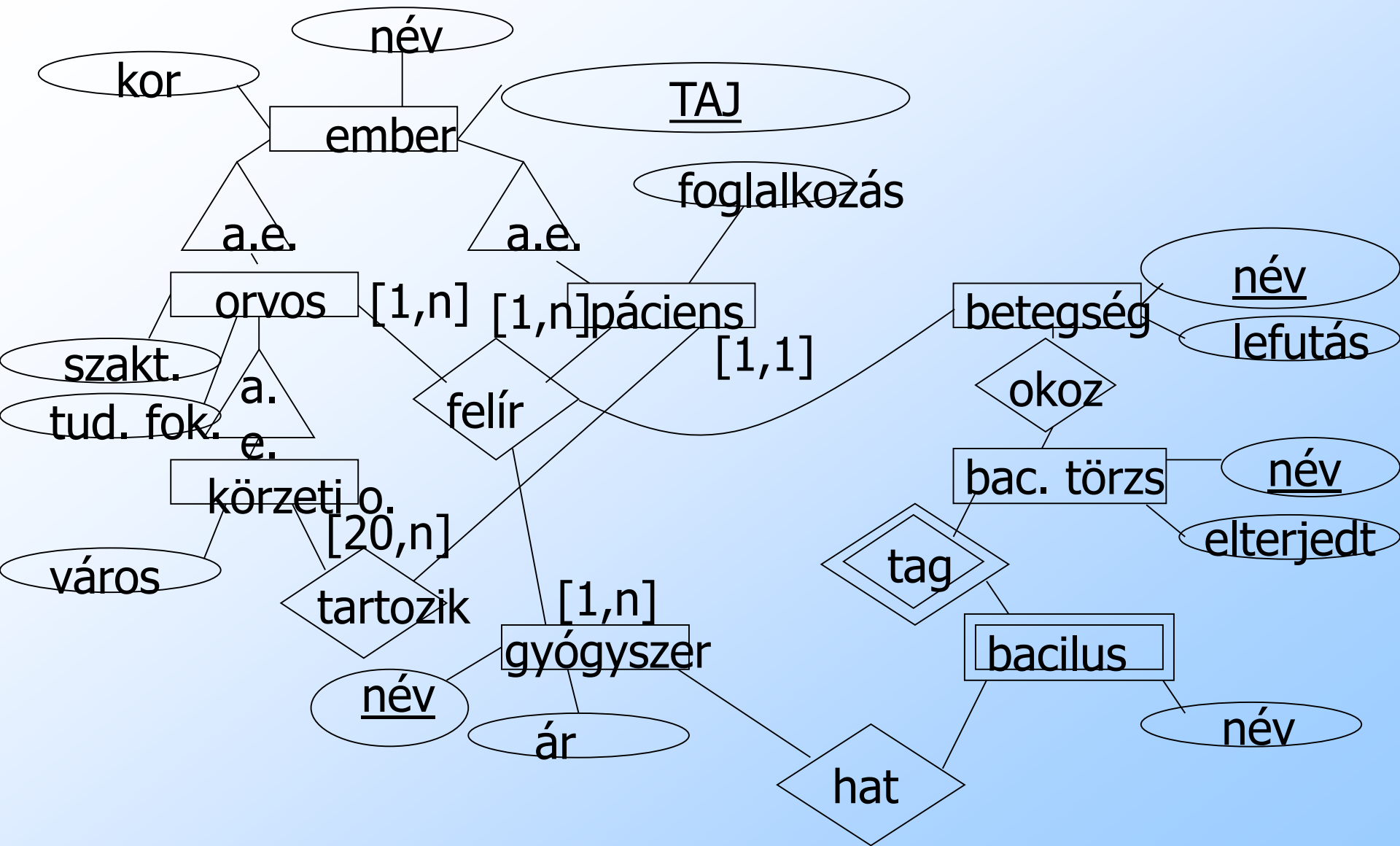












Adatbázis-séma az összevonások előtt

- ◆ Ember (név, kor, TAJ)
- ◆ Orvos (TAJ, szakt., tud_fok)
- ◆ Körzeti orvos (TAJ, város)
- ◆ Páciens (TAJ, foglalkozás)
- ◆ Gyógyszer (név, ár)
- ◆ Betegség (név, lefutás)
- ◆ Bacilus törzs (név, elterjedtség)
- ◆ Bacilus (név, törzs_név)
- ◆ Felír (orvos_TAJ, páciens_TAJ, gy_név, betegség_név)
- ◆ Tartozik (orvos_TAJ, páciens_TAJ)
- ◆ Hatásos (gy_név, bac_név, törzs_név)
- ◆ Okoz (bet_név, törzs_név)

Egyszerűsítés

- ◆ Ember (név, kor, TAJ)
- ◆ Orvos (TAJ, szakt., tud_fok)
- ◆ Körzeti orvos (TAJ, város)
- ◆ Páciens (TAJ, foglalkozás, orvos_TAJ)
- ◆ Gyógyszer (ár, név)
- ◆ Betegség (név, lefutás, törzs_név)
- ◆ Bacilus törzs (név, elterjedtség)
- ◆ Bacilus (név, törzs_név, gy_név) (esetleg)
- ◆ Felír (orvos_TAJ, páciens_TAJ, gy_név, betegség_név)