

# Adatbázisok elmélete

## Egyed-kapcsolat modell

Katona Gyula Y.

Számítástudományi és Információelméleti Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem

- Célja: a modellezendő valóságdarabhoz adatbázisséma létrehozása.
- Elvárás: jól írja le a valóságot, könnyű legyen a gyakori kérdéseket és módosításokat megtenni
- Részei:
  - 1 Terv készítése (nagyon fontos rész, ha rossz tervet csinálunk, később nehéz módosítani) valamilyen modellező eszköz/nyelv segítségével (pl. E/K diagram).
  - 2 A terv átalakítása formálisabb leírássá (tipikusan E/K-ból relációs séma megadása).
  - 3 Az adatbázisséma formális megadása a rendszer által kívánt DDL-en (ez az átalakítás már viszonylag automatikusan megy, a DDL persze rendszerfüggő).

- Először az első lépéssel foglalkozunk, a tervezéssel: E-K modell (egyed-kapcsolat modell). Ez egy grafikus modell a séma megtervezésére.
- Közben végig lesz majd arról szó, hogy hogyan kell a grafikus tervet átírni relációs sémára: ez még mindig egy rendszertől független, elméleti séma lesz (de már nem grafikus)
- Aztán pedig majd lesz az, hogy az SQL DDL-jével hogyan kell megvalósítani a relációs sémát: ez egy konkrét adatbáziskezelő rendszerben levő megvalósítás, az adattáblák létrehozása és a megkötések rögzítése

## Egyed-kapcsolat modell

Adatmodellező eszköz, azaz többé-kevésbé formális jelölésrendszer, adatok és a köztük levő kapcsolatok megadására.

Vannak más modellező eszközök is, de ez a legelterjedtebb.

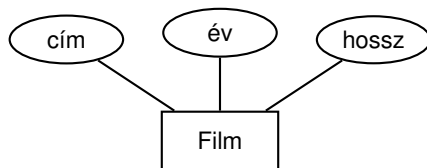
Angolul E/R, azaz entity-relationship modell.

Szemléletes, könnyű vele dolgozni. Egy rajzot készítünk, ez ábrázolja az adatelemeket és a köztük levő kapcsolatokat is.

- *egyedhalmazok*: ezeket akarjuk tárolni, illetve ezekről dolgokat, (pl. pilóta, utas, járat), elemei (példányai) az egyedek (pl. konkrét pilóták, utasok és járatok)
- az egyedhalmazokhoz *attribútumok* tartoznak, ezeket a tulajdonságokat fogjuk az egyedhalmaz minden egyedéről tárolni (pl. pilóta neve, rangfokozata, fizetése, stb.)
- *kapcsolatok*: az egyedhalmazok közötti kapcsolatok, pl. járat utasai, járat személyzete

## Egyedhalmaz attribútumokkal, ábrázolás

Film(cím, gyártási\_év, hossz, ...), rajzon:

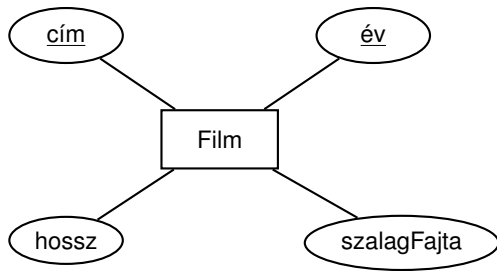


## Egyedhalmaz kulcsa

Olyan attribútum vagy attribútumhalmaz, ami az egyedet egyértelműen azonosítja. (Pl. hallgatónál Neptun-kód vagy filmnél (gyártási év, cím) pár).

Egy kulcsot aláhúzással jelölünk (a kulcsba tartozó attribútumokat aláhúzzuk), ha több kulcs is van, akkor azt az ábrán nem lehet jelölni, ezeket szövegesen mellékeljük

## E/K modell, kulcs ábrázolása



Film(cím, év, hossz, szalagFajta)



## Egyedhalmaz kulcsa

A tervezéskor döntjük el, hogy mik alkossanak kulcsot (persze a valóságot szem előtt tartva). A kulcshoz tartozó attribútumoknak értékeket adva, legfeljebb egy egyed létezhet, amihez ezek az értékek tartoznak.

Néha tűnhet úgy az aktuális adatokból, hogy valami kulcs, mert akkor éppen nincs két egyed ugyanolyan értékekkel, (pl. a film címe kulcsnak tűnhet, amíg nem csinálnak remake-et semmiből), de ettől meg nem lesz kulcs valami, az csak a deklarációtól függ.

## Kis kitérő: relációs adatmodell

Legfontosabb és leggyakoribb a létező adatmodellek közül.

Most: hogyan kell erre átírni az E/K modellt.

Később:

- 1 alpműveletek, elvi keret
- 2 konkrét nyelv: SQL (sémadefinícióra, adatmódosításra és lekérdezésre)
- 3 tervezés: minél jobb séma kialakítása, matematikai elmélet

## Relációs adatmodell

Egyetlen alapfogalom (nincs külön egyedhalmaz és kapcsolat): reláció.

Leginkább úgy gondolunk a relációra, mint egy síkbeli táblázatra:

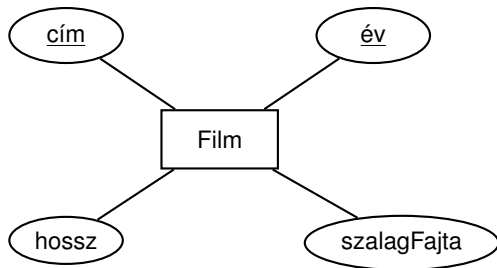
$R_1$	$A_1$	$A_2$
	1	y
	1	z
	3	y

$R_2$	$A_1$	$A_3$
	2	y
	1	z

Itt  $R_1$  a reláció neve,  $A_1$  és  $A_2$  az attribútumok nevei, a sorok pedig a reláció elemei. Az oszlopokban levő értékek az attribútumokhoz tartozó értékészletről kerülnek ki.

## E/K modell átírása relációsra, alapeset

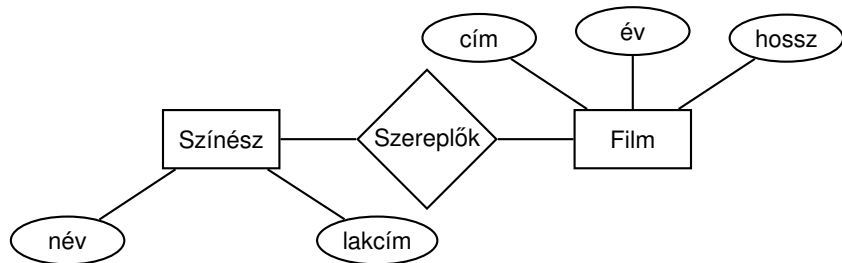
Egyedhalmaz attribútumokkal:



Film(cím, év, hossz, szalagFajta)

A reláció kulcsa = az egyedhalmaz kulcsa

Szereplők(Film, Színész):



## Egy kapcsolat példányai

Ha az  $R(E_1, E_2, \dots, E_{10})$  kapcsolat 10 egyedhalmazt köt össze, akkor az  $R$  kapcsolat egy példánya egy 10 hosszú vektor  $(e_1, e_2, \dots, e_{10})$ , ahol az  $e_i$  egy egyed az  $E_i$  egyedhalmazból.

Például Szereplők(Film, Színész) kapcsolat esetén példányok:

(Éhezők viadala, Jennifer Lawrence)

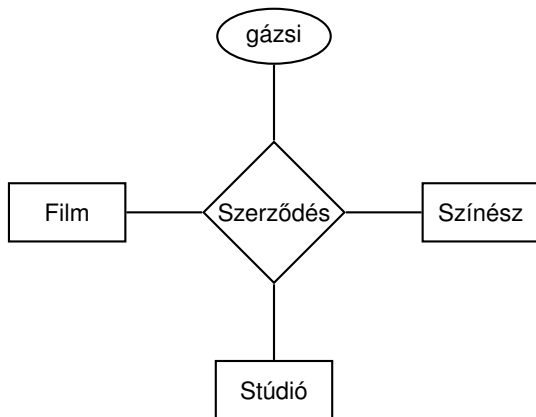
(Éhezők viadala, Elizabeth Banks)

(Napos oldal, Jennifer Lawrence)

(Testről és lélekről, Borbély Alexandra)

## Kapcsolat attribútuma

Az E/K modellben a kapcsolatnak is lehet attribútuma (ez nem minden modellben van így):



Itt a gázsi a szerződéshez tartozik, ami a filmet, a színészt és a stúdiót köti össze.

## Kapcsolatok típusa bináris kapcsolat esetén

- $R(E_1, E_2)$  több-több (many-many) kapcsolat: egy  $E_1$ -beli egyedhez több  $E_2$ -beli tartozhat és egy  $E_2$ -beli egyedhez több  $E_1$ -beli tartozhat,  
például Szerepel(Film, Színész)
- $R(E_1, E_2)$   $E_2$  irányba több-egy (many-one) kapcsolat: egy  $E_1$ -beli egyedhez csak egy  $E_2$ -beli tartozhat, de egy  $E_2$ -beli egyedhez több  $E_1$ -beli is tartozhat,  
például Anyja(Személy, Személy)
- $R(E_1, E_2)$  egy-egy (one-one) kapcsolat: egy  $E_1$ -beli egyedhez csak egy  $E_2$ -beli tartozhat és egy  $E_2$ -beli egyedhez is csak egy  $E_1$ -beli is tartozhat,  
például Házastársa(Személy, Személy)

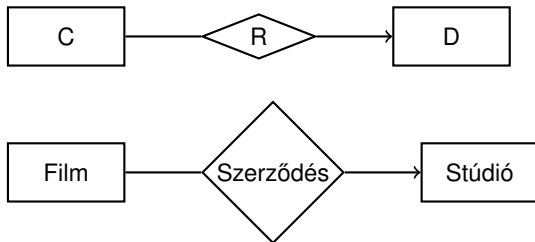


- Fontos: a kapcsolat típusa modellezési kérdés, azt mutatja, hogy mit gondolunk a világról
- Ha egy kapcsolatot valamelyik irányba „egy”-nek jelölünk, akkor az azt jelenti, hogy a relációs modellben majd lesz egy megkötés, ami ezt kikényszeríti.
- Az, hogy egy kapcsolat valamelyik irányba „több”, az csak egy lehetőség: tartozhat egy Színészhez több Film is, de persze lehetnek elsőfilmes színészek is.

## Kapcsolatok típusának ábrázolása bináris kapcsolat esetén

Az „egy” irányt nyíl jelzi, azaz ott van nyíl, amelyik egyedhalmazból csak egy tartozhat a másik egyedhalmaz egy egyedéhez.

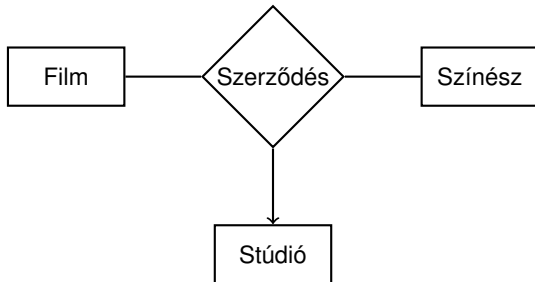
Például, ha a  $C$  egyedhalmaz egy egyedéhez csak egy  $D$ -beli tartozhat az  $R$  kapcsolatnál, de egy  $D$ -belihez tartozhat több  $C$ -beli is, akkor:



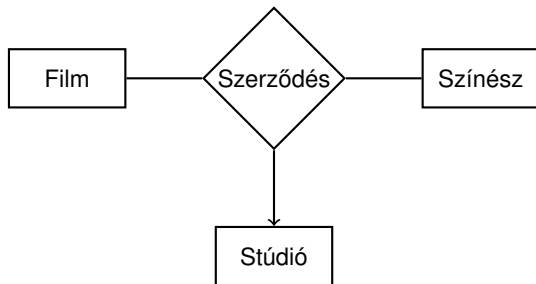
Ha egy-egy kapcsolat van, akkor persze mindkét oldalra kell a nyíl.

## Többágú kapcsolat típusa

Az  $R(E_1, E_2, \dots, E_n)$  kapcsolat  $E_i$  felé egyirányú, ha igaz az, hogy a maradék  $E_1, E_2, \dots, E_{i-1}, E_{i+1}, \dots, E_n$  egyedhalmazokból bárhogy választva ki egy-egy egyedet, maximum egy olyan  $E_i$ -beli  $e_i$  egyed van, amivel az  $(e_1, e_2, \dots, e_{i-1}, e_i, e_{i+1}, \dots, e_n)$  egyedvektorra az  $R$  kapcsolat fennáll. (Természetesen lehet egy többágú kapcsolat egynél több irányban is „egy” jellegű.) Példa:



## Többágú kapcsolatra példa

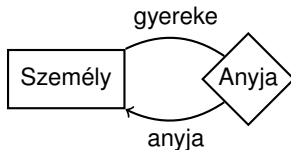


Egy rögzített (film, színész) párhoz csak egy stúdió tartozik, de pl. egy rögzített (stúdió, film) párhoz tartozhat több színész.

Nem minden írható le ilyen módon, de nem baj, úgymint az a fontos, hogy majd a relációs megadásnál pontosak tudjunk lenni.

## Kapcsolatok ábrázolása, szerepek

Ha egy kapcsolatban egy egyedhalmaz többször is szerepel, akkor a nyilakon/vonalakon jelöljük a különböző szerepeket. pl:



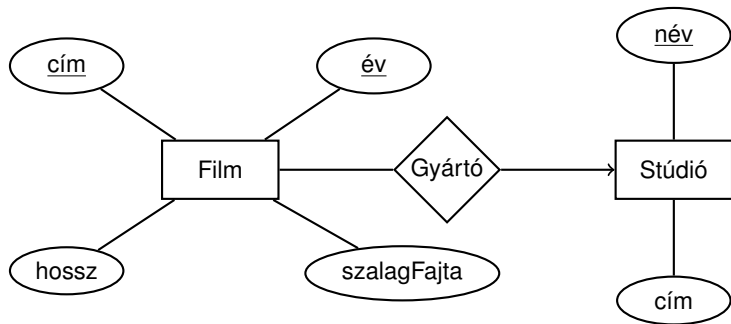
Fontos megcímkézni a vonalakat, mert ahhoz a szerephez tartozik az „egy” nyíl, ami az anyára vonatkozik, a gyerekághoz nem kell.

## Kapcsolat átírása relációvá

Minden kapcsolatból külön reláció, attribútumai: a kapcsolatban résztvevő egyedhalmazok kulcsainak uniója + kapcsolat attribútumai (esetleg átnevezés)

Az így kapott reláció kulcsa: a kapcsolatban résztvevő egyedhalmazok kulcsainak uniója

## Kapcsolat átírása relációvá



Gyártó(cím, év, név)

Film(cím, év, hossz, szalagFajta)

Stúdiók(név, cím)

## Kapcsolat átírása relációvá, speciális eset

Ha bináris több-egy kapcsolatról van szó, akkor van jobb megoldás is:

Nem veszünk fel külön relációt a kapcsolatnak, hanem ha az  $E$  és  $F$  közti kapcsolat  $F$  felé egyirányú, akkor az  $E$  egyedhalmaz átírásakor bevesszük az  $F$  osztály kulcsát is.

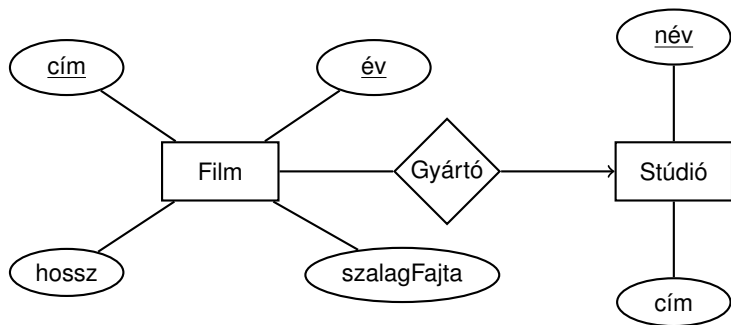
Miért jó?

- az  $E$ -beli kulcs meghatározza az  $E$ -beli egyedet, az pedig meghatározza az  $F$ -belit
- eggyel kevesebb tábla lesz
- mivel egy  $E$ -belihez csak egy  $F$ -beli tartozik, ezért nem lesz redundáns

Az  $E$ -hez tartozó reláció kulcsa  $E$  kulcsával egyezik meg.



## Kapcsolat átírása relációvá, speciális eset



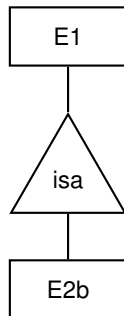
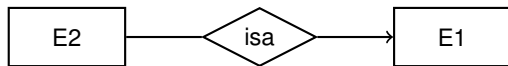
Így most nem kell külön tábla a kapcsolatnak, hanem a Film(cím, év, hossz, szalagFajta, **név**) lesz a Film tábla.

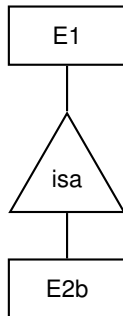
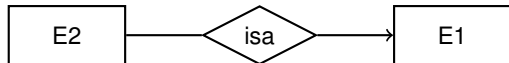
## Alosztályok

Egy egyedhalmaz speciális tulajdonságú, de egymáshoz hasonló egyedei alkotják. Ezeket érdemes együtt kezelni, de úgy, hogy a (nagyobb) egyedhalmazba való tartozásuk is megmaradjon.

Ehhez egy speciális (*isa*, magyarul *azegy*) kapcsolat van. (Motiváció: A student is a person = a diákok alosztályát alkotják az embereknek)

Jelölés, ha  $E_2$  alosztálya  $E_1$ -nek:

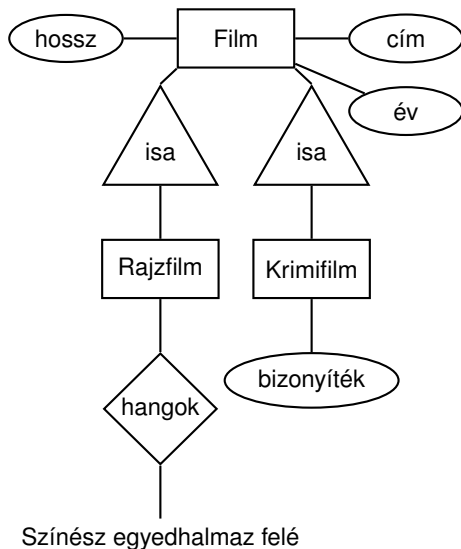




Az első esetben a nyíl arra mutat, amelyik a fő osztály, ez most nem a több-egy kapcsolatnál megszokott nyíl, az *isa* ilyen szempontból is speciális kapcsolat. A második esetben az alosztály van alul. Az alárendelt halmaz örökli a fő halmaz attribútumait és kapcsolatait, de persze lehetnek neki sajátjai is.

## Alosztályokra még egy példa

A Film egyedhalmazon belül akarhatunk külön Rajzfilm, Vígjáték, Krimifilm kategóriákat:



## Alosztályokra még egy példa

Ha egy egyed több alosztályba is tartozik, akkor az attribútumait/kapcsolatait a felmenőtől szedi össze. Így az E/K modellben egy olyan film, ami krimi is és rajzfilm is, három helyről szedi össze az attribútumait/kapcsolatait: a címét, hosszát és gyártási évét a Film egyedhalmazból, a hangjait a Rajzfilm alosztályból, a bizonyítékot pedig a Krimifilmből.

A relációs modellre való átíráskor majd ezt ügyesen kezeljük.

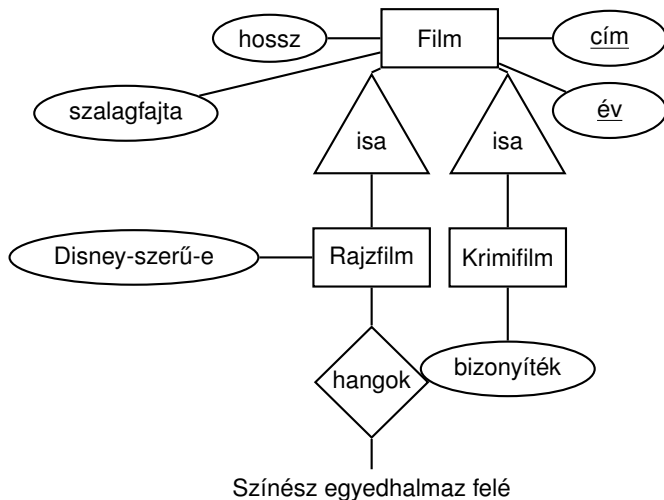
## Alosztályok kezelése a relációsra való átíráskor

Mivel E/K-ban egy egyed lehet több egyedhalmazban is lehet egyszerre, ezért az egy filmre vonatkozó információk szét vannak szórva. A relációs sémára való átíráskor gondoskodunk róla, hogy a részinfókból vissza tudjuk állítani az egészet (ha kell).

Első megoldás:

Minden alosztályhoz a főosztály kulcsát és saját attribútumait rendeljük. Az alosztály kulcs a főosztály kulcsa lesz, így a kapcsolataiba is ezt viszi magával az alosztály. Az „isa” kapcsolathoz nem rendelünk relációt.

## Alosztályok kezelése a relációsra való átíráskor

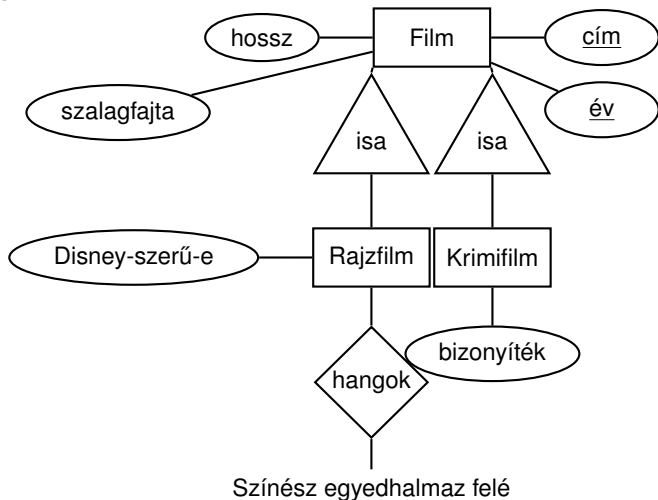


Film(cím, év, hossz, szalagfajta),  
Rajzfilm(cím, év, Disney-szerű-e),  
Krimi(cím, év, bizonyíték),  
Hangok(cím, év, Színésznév)

Ennek a módszernek a hátránya: egy film információi több helyre vannak szórva.  
Pl. Macskafogónál: a hossz és a szalagfajta a Film-ben, az, hogy nem Disney-is, az a Rajzfilmben, hangok a Hangokban. De ezeket az infókat össze lehet rakni, a (cím, év) kulcs menti természetes illesztéssel.



## Másik megoldás NULL értékkel



Film(cím, év, hossz, szalagfajta, Disney-szerű-e, bizonyíték)

A hiányzó helyeket NULL-al töltjük ki (pl. NULL van a Disney-szerű-nél ott, ami nem rajzfilm)

## Másik megoldás NULL értékkel

Hátrányok:

- 1 elveszíthetünk információt. Pl. egy olyan krimiről, amiben nincs bizonyíték, nem tudjuk, hogy krimi, illetve ha egy rajzfilmről nem tudjuk, hogy Disney-s vagy sem, akkor elvesztjük azt az infót is, hogy rajzfilm
- 2 NULL értékekre nagyon kell figyelni

## NULL értékek, ízelítő

Ez a két SQL kérdés vajon ugyanaz?

```
SELECT * FROM Dolgozó WHERE (fizetés < 400 000 ) OR ( Fizetés ≥ 400 000);
```

```
SELECT * FROM Dolgozó;
```

Nem! Erről majd bővebben SQL-nél, most csak annyi a lényeg, hogy résen kell lenni, ha NULL-ról van szó.

Olyan megszorításokat is ábrázolni akarunk amik nem fejezhetők ki pusztán az attribútumok és a kapcsolatok felsorolásával vagy a kapcsolatok típusával (pl. több-egy).

Ezek további infók, amik

- a séma részei, ezért már a tervezéskor kell rájuk gondolni,
- olyan megkötéseket tartalmaznak, amikre majd mindig figyelni kell.

Nem világos, hogy mik a jó megkötések, miket lehet jól kezelni.

## Tipikus megszorítások

*Kulcs*: olyan attribútum, vagy attribútumhalmaz megadása, ami az egyedet egyértelműen azonosítja. (Pl. hallgatónál Neptun-kód vagy filmnél (gyártási év, cím, stúdió) hármassal).

A tervezéskor döntjük el, hogy mik alkossanak kulcsot (persze a valóságot szem előtt tartva). A kulcshoz tartozó attribútumoknak értékeket adva, legfeljebb egy egyed létezhet, amihez ezek az értékek tartoznak.

Néha tűnhet úgy az aktuális adatokból, hogy valami kulcs (mert akkor éppen nincs két egyed ugyanolyan értékekkel, (pl. a film címe kulcsnak tűnhet, amíg nem csinálnak remake-et semmiből), de ettől meg nem lesz kulcs valami, az csak a deklarációtól függ.

## Tipikus megszorítások

- *Egyértékűségi megszorítások*: előírhatjuk, hogy valami érték vagy érték kombináció legyen egyedi. Pl. ilyen a kulcsok megadása is, vagy az, hogy bizonyos attribútumok értékei meghatározzák más attribútumok értékeit (ezek lesznek majd a funkcionális függések a relációs modellben, erről később lesz szó).

Például ilyen az, hogy az irányítószám meghatározza a várost

- *Hivatkozási épség*: a hivatkozott dolognak léteznie kell, pl. ha egy tárgynál szerepel egy hallgató, akkor az a hallgató a hallgatók alapadatait tartalmazó táblában is benne legyen

## Tipikus megszorítások

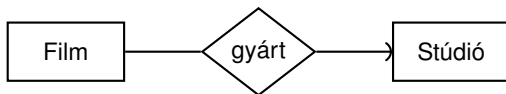
- *Értelmezési tartomány korlátozása*: attribútum lehetséges értékeire megkötés (pl. magasság legyen kisebb 300-nál, film gyártási éve 1800 utáni). Módszerek erre: konkrétan majd az SQL DDL-jénél.
- *Egyéb megszorítások*: minden más, pl. kapcsolat fokának korlátozása (egy filmnek max. 10 szereplőjét akarjuk nyilvántartani).

- Alapkérdés: Mik a jó megszorítások? Miket lehet megvalósítani? Ez persze majd a konkrét megvalósítástól függ, a konkrét DDL-től, de azért jó lenne már a modellezéskor is annyit leírni, amennyit csak lehet, aztán majd meglátjuk.
- A megszorítások haszna
  - ▶ jobban/valósághoz közelebbi módon le lehet velük írni a világot
  - ▶ segíthetik a tárolást (elég pl. a kulcsattribútumokat megadni kereséskor)



## Megszorítások ábrázolása E/K modellben

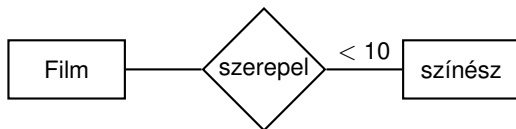
- Elsődleges kulcs aláhúzással, többi szövegesen
- Hivatkozási épség: lehet a rajzon jelezni, ha egy kapcsolatnál azt szeretnénk, hogy pontosan egy egyed tartozzon egy kiválasztott egyedhez. Ilyenkor kerek nyilat használunk:



Ebben az esetben minden filmhez pontosan egy stúdiónak kell tartoznia.

## Megszorítások ábrázolása E/K modellben

Kapcsolat fokát lehet korlátozni, pl:



Ekkor egy filmhez 10-nél kevesebb színészt rendelünk.

## Megszorítások átírása relációsra

Erről majd később lesz szó, amikor a relációs modell elméleti kereteit tanuljuk (funkcionális függések), illetve amikor az SQL DDL-jét tanuljuk.

Most elégedjünk meg annyival, hogy a kulcsokat át tudjuk valahogyan írni, a többről meg azt képzeljük, hogy minden, az E/K digramon jelölt megszorítás szövegesen van leírva és majd meglátjuk, hogy hogyan lehet ezeket implementálni.

## Gyenge egyedhalmazok

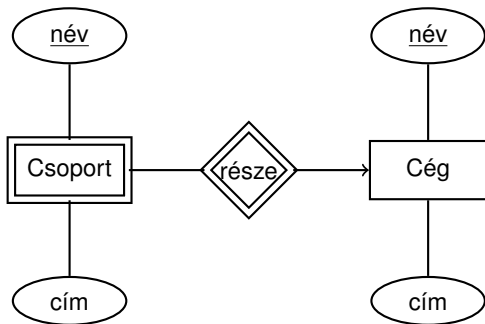
Általában azt gondoljuk, hogy a valóságban egy egyedet az összes attribútumának értékét megadva azonosítunk, de ez nem mindig van így.

Egy egyedhalmaz akkor gyenge egyedhalmaz, ha az egyedeit nem azonosítják az attribútumai magában, csak a kapcsolatokkal együtt.

Jelölés: dupla téglalap az egyedhalmaznak és dupla rombusz azoknak a kapcsolatoknak, amiken keresztül megy az azonosítás.

A gyenge egyedhalmaznál az aláhúzott attribútumok belekerülnek a gyenge egyedhalmaz kulcsába, de még más attribútumok is hozzájönnek ehhez: azok, amik a duplarombuszos kapcsolat(ok) végén álló egyedhalmaz(ok) kulcsai.

## Gyenge egyedhalmazok, példa



Ebben a példában a csoport neve még önmagában nem kulcs (sok cégnél lehet pl. HR csoport), sőt a címmel együtt sem feltétlenül azonosít egy csoportot, de ha a kapcsolaton keresztül a céget is be vesszük az azonosításba, úgy már egyértelmű lesz, hogy melyik csoportról beszélünk.

## Gyenge egyedhalmazok: követelmények az azonosító kapcsolatra

A gyenge egyedhalmaz kulcsában benne lehetnek saját attribútumai (mint az előbb a Csoport neve) és biztosan vannak benne olyan attribútumok, amiket duplarombuszos kapcsolat(ok)on keresztül szerez.

Követelmények ezekre a kapcsolatokra:

- 1 Ha az  $E$  gyenge egyedhalmaz kulcsattribútumot szerez egy  $F$  egyedhalmaztól az  $R$  kapcsolaton át, akkor  $R$  több-egy  $E$ -ből  $F$ -be. (Így egy  $E$ -belihez egyértelműen tartozik egy  $F$ -beli.)
- 2 Egy  $F$ -beli attribútum pontosan akkor kerül bele az  $E$  gyenge egyedhalmaz kulcsába, ha benne van az  $F$  egyedhalmaz kulcsában is.

Megjegyzés: természetesen  $F$  is lehet gyenge egyedhalmaz.

## Miért vannak gyenge egyedhalmazok?

Leginkább a redundancia elkerülése céljából. (Minek a cég nevét minden csoportnál külön felvenni, elég ha egyszer felírjuk és a kapcsolatból derítjük ki.)

A redundancia elkerülése nem csak az E/K modellben fontos, ez minden megközelítésben lényeges, hisz a redundancia bajok forrása:

- Nehéz konzisztens állapotban tartani az adatbázist, ha ugyanaz az infó ezer helyen van beírva.
- Helyprobléma.

Törekszünk a redundancia kiküszöbölésére, de persze nem kell mindent kiirtani, hisz a világ is redundáns.

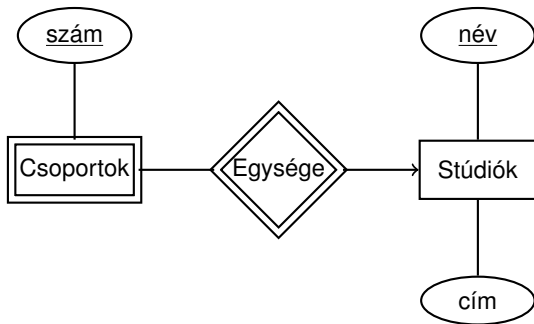
## Gyenge egyedhalmazok átírása relációra

Ha  $W$  gyenge egyedhalmaz:

- Az egyedhalmazból reláció lesz persze
- De a relációbak nem csak  $W$  attribútumait kell tartalmaznia, hanem azokat is, amiktől kulcs lesz. (Dupla keretes kapcsolat.)
- Ez minden olyan kapcsolatra is igaz, melyben  $W$  részt vesz és amelyben így szerepel  $W$  kulcsa.
- A dupla keretes kapcsolatokhoz általában nem kell külön reláció (mert az az infó már egyszer szerepel a gyenge egyedhalmaz megadásánál).



## Példa

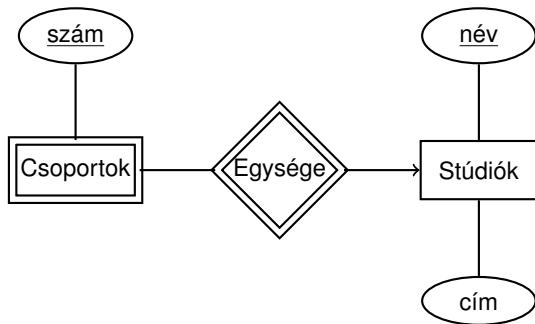


Stúdió(név, cím)

Csoportok(szám, név)

Egysége(szám, név, név)

## Példa



Egysége (szám, név, név) helyett Egysége (szám, név) hiszen ugyanaz kétszer.

Sőt!

Egysége el is hagyható, hiszen összes attribútuma szerepel a Csoport-ban is.

## Megjegyzés: sokágú kapcsolat átírása binárisra az E/K modellen belül

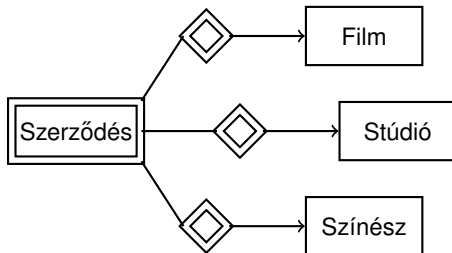
Miért lehet jó ez?

- átláthatóbb a bináris
- a megvalósításban könnyebben kezelhető

Az átíráshoz fő ötlet: egy  $n$  ágú  $R(E_1, E_2, \dots, E_n)$  kapcsolat egy eleme egy  $n$  egyedből álló vektor :  $(e_1, e_2, \dots, e_n)$ . Az átírásnál létrehozunk egy új (gyenge) egyedhalmazt, melynek az ilyen vektorok lesznek az egyedei és az ilyen vektorokat bináris több-egy kapcsolatok ( $n$  darab) fogják az  $E_1, \dots, E_n$  egyedhalmazokhoz kötni.

## Sokágú kapcsolat átírása binárisra

A háromágú szerződéses példa átírva:



## Megjegyzések a binárisra való átíráshoz

- A relációsra való átírásnál a többes kapcsolatot reprezentáló reláció egy sora úgyis éppen ez lenne, ami itt a gyenge egyedhalmaz egy eleme (ugyanott tartunk majd a végén).
- Az átírásnál persze veszíthetünk infót: az előbbi példánál elveszett az, hogy a többes kapcsolat „egy” volt a Stúdió fele. Ez nem baj, ezt majd a végén a relációs modellben úgyis finomabban le tudjuk írni a funkcionális függésekkel.

- 1 *Valóság-hű modellezés*: megragadni a lényegét, megfelelő adatelemeket választani, megfelelő kapcsolatokat
- 2 *Redundancia kerülése* észszerű mértékben. Ezt majd a relációs modell nagyon jól megoldja, de azért már a tervezéskor is jó erre figyelni.
- 3 *Egyszerűség*: csak az legyen a sémában, aminek lennie kell, minél egyszerűbb szerkezetben.

- 4 *Megfelelő (típusú, összetettségű) adatelemek választása:* jól döntsünk, hogy mi legyen attribútum, mi inkább kapcsolat, illetve esetleg külön osztály/egyedhalmaz. Az attribútumot egyszerűbb implementálni, de néha átláthatóbb egy külön egyedhalmaz.

### Általános elvek:

- ▶ ha egy egyedhalmaznak csak egy attribútuma lenne, akkor nem érdemes külön venni, ha összetettebb, akkor legyen külön
- ▶ ha egy infót magában nem akarunk megőrizni, csak valamihez kapcsolatva, akkor legyen csak attribútum (pl. ha a stúdiók csak annyiban érdekelnek minket, hogy melyik filmet ki gyártja, akkor nem kell külön Stúdió egyedhalmaz)