Adatbázisok elmélete 8. előadás

Csima Judit
Budapesti Műszaki és Gazdaságtudományi Egyetem
Számítástudományi Tsz.
I. B. 136/b
csima@cs.bme.hu

2003. Március 11.

Kapcsolatok átírása

Ha valamelyik irányba egyértékű a kapcsolat: ha a C és D közti kapcsolat D felé egyirányú, akkor a C osztály átírásakor bevesszük a D osztály kulcsát is (ha több van, akkor egyet)
Csak egyik irányból csináljuk, akkor is, ha a másik irányba is "egy" a kapcsolat

```
interface Film (keys (cím, év)){
    attribute cím;
    attribute év;
    attribute hossz;
    attribute szalagFajta;
    relationship Stúdió gyártó
        inverse Stúdió::gyárt; };
```

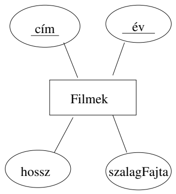
$\Longrightarrow$ Film(<u>cím</u>, <u>év</u>, hossz, szalagFajta, **stúdióNév**)
Feltéve, hogy a stúdiónév kulcs a Stúdió osztályban

Kulcs: mivel a kapcsolat "egy" jellegű volt, ezért az osztály kulcsa jó lesz kulcsnak a relációsémában is

Mindkét irányban többértékű kapcsolat: Ugyanaz a probléma, mint a halmaz típusú attribútum. Nem lehet jól megoldani, sok sor lesz és a kulcs is elromlik. Ha több ilyen kapcsolat is van $\Longrightarrow$ katasztrófa

E/K modell átírása

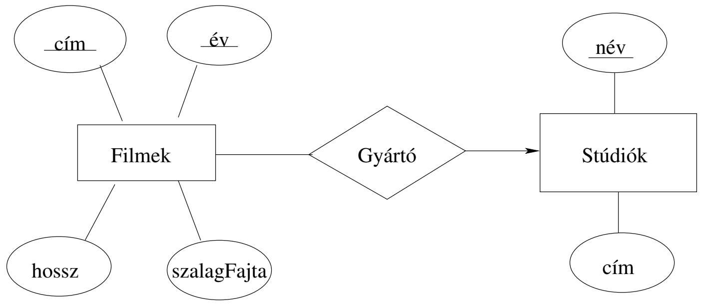
Egyedhalmaz attribútumokkal:



$\Longrightarrow$ Film(<u>cím</u>, <u>év</u>, hossz, szalagFajta)

A reláció kulcsa = az egyedhalmaz kulcsa

Egyértékű és többértékű kapcsolat:
$\Longrightarrow$ Külön reláció, attribútumai: a kapcsolatban résztvevő egyedhalmaz kulcsainak uniója + kapcsolat attribútumai (esetleg átnevezés)
Az így kapott reláció kulcsa: a kapcsolatban résztvevő egyedhalmaz kulcsainak uniója



$\Longrightarrow$ `Gyártó(cím, év, stúdióNév)`
`Film(cím, év, hossz, szalagFajta)`
`Stúdiók(stúdióNév, cím)`

E/K-ból jobban lehet relációsat csinálni.

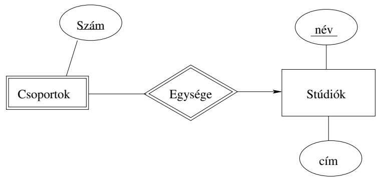
Megjegyzés: ha bináris több-egy kapcsolatról van szó, akkor van jobb megoldás is, az amit ODL-nél csináltunk:
ha az E és F közti kapcsolat F felé egyirányú, akkor az E egyedhalmaz átírásakor bevesszük az F osztály kulcsát is. Ez ugyanazért lesz jó, miért az ODL-es és így eggyel kevesebb tábla lesz.

Így az előbbi E/K diagram esetén nem kell külön tábla a kapcsolatnak, hanem a Film(cím, év, hossz, szalagFajta, **stúdióNév**)
lesz a Film tábla.

Gyenge egyedhalmazok kezelése

Ha W gyenge egyedhalmaz:

- 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 nem kell külön reláció (mert az az infó már egyszer szerepel a gyenge egyedhalmaz megadásánál).



`Stúdió(név, cím)`
`Csoportok(szám, stúdióNév)`
`Egység(szám, stúdióNév, név)`

`Egység(szám, stúdióNév, név)` $\Longrightarrow$ `Egység(szám, név)` hiszen ugyanaz kétszer.
$\Longrightarrow$ `Egység` el is hagyható, hiszen összes attribútuma szerepel a `Csoport`-ban is.
Ez általában is igaz, hiszen a gyenge egyedhalmaz attribútumai között ott lesz a dupla keretes rombusz minden attribútuma.

Alosztályok kezelése ODL-ben

`Film`, `Rajzfilm`, `Krimi`, `KrimiRajzfilm` esete. *(Itt minden egyed pont egy osztályban lehet benne, ezért kellett KrimiRajzfilm osztályt is megadni, pl. a Macskafogó miatt.)*

Minden alosztályhoz egy reláció, minden attribútumával és kapcsolatával (öröklöttekkel is).

```
Film(cím, év, hossz, szalagFajta, stúdióNév, színésznév)
Rajzfilm(cím, év, hossz, szalagFajta, stúdióNév, színésznév, hang)
BűnügyiFilm(cím, év, hossz, szalagFajta, stúdióNév, színésznév,
fegyver)
BűnügyiRajzfilm(cím, év, hossz, szalagFajta, stúdióNév, színésznév,
hang, fegyver)
```

Kulcs: a főosztálynál úgy, ahogy eddig volt, az alosztály meg örökli a kulcsot, ha tudja

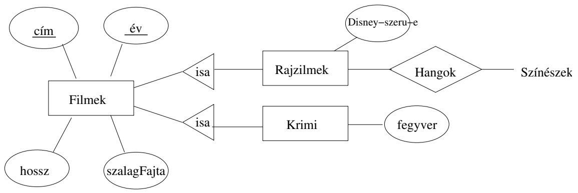
Hátrány: egy film kereséséhez mind a négy relációt végig kell nézni, ha nem tudjuk, hogy hova tartozik a keresett film.

Megjegyzés: ebben a példában a kulcsok elromlanak a többes kapcsolatok miatt

Alosztályok kezelése E/K modellben

E/K-ban nem kell egy egyednek egyetlen egyedhalmazban lennie (ezért nem lesz itt KrimiRajzfilm egyedhalmaz), előfordulhat, hogy egy filmre vonatkozó információk szét vannak szórva. A relációs sémára való átíráskor gondoskodunk róla, hogy a részinfókból vissza tudjuk állítani az egészet.

Átírás: Minden alosztályhoz csak a főosztály kulcsát és saját attribútumait rendeljük.
Az alosztály kulcs a főosztály kulcsa lesz, így a kapcsolatba ist ezt viszi magával az alosztály.
Az "isa" kapcsolathoz nem rendelünk relációt.



$\Longrightarrow$

```
Film(cím, év, hossz, szalagFajta)
Rajzfilm(cím, év, Disney-szerű-e)
Krimi(cím, év, fegyver)
Hangok(cím, év, Szinésznév)
```

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, stúdióNév, színésznév, fegyver)`

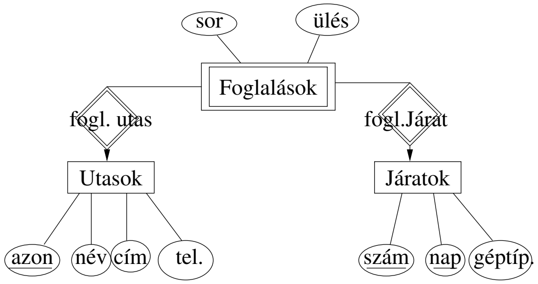
A hiányzó helyeket NULL-al töltjük ki.

Hátrány:

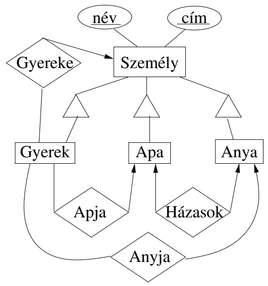
1. elveszíthetünk információt. Pl. egy olyan krimiről, amiben nincs fegyver, nem tudjuk, hogy krimi
2. a (cím, év) pár nem lesz kulcs, ugyanúgy, ahogy az ODL-es átírásnál sem lett

Példák

Alakítsuk relációssá:




```
Utas(azon, név, cím, tel.)
Járat(szám, nap, géptípus)
Foglalások(azon,szám, nap, sor, ülés)
```


```
Személy(név, cím); Gyerek(név, cím); Apa(név, cím)
Anya(név, cím)
Apja(apanév, apacím, gyereknév, gyerekcím)
Anyja(anyanév, anyacím, gyereknév, gyerekcím)
Házast(fnév, fcím, nőnév, nőcím)
Gyereke(gyereknév, gyerekcím, szülőnév, szülőcím)
```


Sorkalkulus

Lekérdezőnyelvek típusai:

- relációs algebra (**LEAP**, letölthető, SIGMOD-ról link),**ISBL** *nehezen emészthetőbb, algebrai alapú; ez volt: láttuk, hogy relációs algebrával jól meg lehet adni relációkat*
- sorkalkulus (**SQL egy kicsit, QUEL**, INGRES nyelve), *könnyebben emészthető, logikai alapú*
- oszlopkalkulus (**QBE, SQL**) nagyon hasonló

Sorkalkulus (Tuple calculus)

Formális modell, de már hasonlít az igazihoz.

Elsőrendű nyelv relációk kifejezésére.

Változók: t, r, s sorváltozók, a reláció sorainak felel meg

$t^{(k)}$: k oszlopos reláció sorainak felel meg

$t^{(k)}[i]$: A t sorváltozó i-edik komponense.

Pl. egy sor $\Longrightarrow$ (R. M., Budapest, hamburger, 180), akkor
$t[3] =$'hamburger' és t[ÁR] $= 180$

Cél: sorkalkulussal relációkat megadni, úgy, mint relációs algebrával

A sorkalkulus által kifejezett reláció:
$\{t^{(k)} \mid \phi(t)\}$ $\Longrightarrow$ a kifejezett reláció azon t-kből áll, amikre $\phi(t)$ igaz, ahol ϕ egy megengedett formula+ valami még.

Megengedett formulák (amik a $\phi(t)$ helyén állhatnak):

atomok :

- $R^{(k)}(t^{(k)})$: *(ahol R alapreláció), akkor igaz, ha $t \in R$, azaz a sor benne van a relációban.*
- – $t^{(k)}[i]\ \theta\ s^{(l)}[j]$

– $t^{(k)}[i]\ \theta\ c$
– $c\ \theta\ t^{(k)}[i]$
ahol $\theta \in \{<, >, =, \neq, \leq, \geq\}$, t, s sorváltozók, c konstans érték.
Világos, mikor igaz.

építkezési szabályok :

- ϕ, ψ formulák, akkor $\phi \vee \psi, \phi \wedge \psi, \neg\phi$ is formulák.
  Világos, hogy mikor igaz.
- ϕ formula, s sorváltozó, akkor $\forall s \phi, \exists s \phi$ is formula.
  Világos, hogy mikor igaz.
  Kötött változó: ha vonatkozik rá kvantor,
  Szabad változó: ha nem,

Sorkalkulus által kifejezett reláció (pontosan) : $\{t^{(k)} \mid \phi(t)\}$ $\Longrightarrow$ a kifejezett reláció azon k hosszú t vektorokból áll,amikre $\phi(t)$ igaz, ahol ϕ egy megengedett formula és ϕ-*ben t az egyetlen szabad* változó.