

Adatbázisok elmélete 6. előadás

Katona Gyula Y.
Budapesti Műszaki és Gazdaságtudományi Egyetem
Számítástudományi Tsz.
I. B. 137/b
kiskat@cs.bme.hu
http://www.cs.bme.hu/~kiskat

2005

Bizonyítás

- **alapreláció:** Tegyük fel, hogy R k oszlopos alapreláció
 $R = \{t^{(k)} \mid R(t)\}$
- **Unió:** Tfh. S is k oszlopos
 $R \cup S = \{t^{(k)} \mid R(t) \vee S(t)\}$
- **különbség:**
 $R \setminus S = \{t^{(k)} \mid R(t) \wedge \neg S(t)\}$
- **metszet:**
 $R \cap S = \{t^{(k)} \mid R(t) \wedge S(t)\}$
- **szorzat:** R legyen k oszlopos, S pedig l oszlopos

$$R \times S = \{t^{(k+l)} \mid \exists r^{(k)} \exists s^{(l)} (R(r) \wedge S(s) \wedge r[1] = t[1] \wedge \dots$$

$$\wedge r[k] = t[k] \wedge s[1] = t[k+1] \wedge \dots \wedge s[l] = t[k+l])\}$$

- **vetület:** Legyen $R(A_1, \dots, A_d, A_{d+1}, \dots, A_k)$ reláció, vetítsük az első d -re
 $\pi_{A_1, \dots, A_d}(R) = \{t^{(d)} \mid \exists r^{(k)} (R(r) \wedge r[1] = t[1] \wedge \dots \wedge r[d] = t[d])\}$

Sorkalkulus ereje

Mivel lehet több mindent kifejezni?

Relációs algebraival, vagy sorkalkulussal?

Tétel. A sorkalkulus relációsan teljes.

Bizonyítás: Be kell látni, hogy minden reláció, ami relációs algebraival megadható, megadható sorkalkulussal is. Ehhez azt elég megmutatni, hogy

1. az alaprelációk megadhatók
2. a relációs algebrai alapműveletek (unió, különbség, szorzat, vetítés, szelekció) alaprelációkra alkalmazva megvalósíthatók
3. ha R és S nem alapreláció és ezekre alkalmazunk valami relációs alapműveletet, akkor az eredmény kifejezhető sorkalkulussal

- **kiválasztás:**
 $\sigma_{F'}(R) = \{t^{(k)} \mid R(t) \wedge F'\}$, ahol F' átfordítása sorkalkulusra \implies az i -edik attribútum helyett $t^{(n)}[i]$ -t írunk.
Pl. (evidenciával történő meggyőzés)
 $\sigma_{\text{'ÁR>'150' \wedge \text{TERMÉK='hamburger' (TERMEL)}} =$
 $\{t^{(4)} \mid \text{TERMEL}(t) \wedge t[4] > '150' \wedge t[3] = \text{'hamburger'}\}$
- **Nem lényeges, hogy R, S alaprelációk.**
Ha $R = \{t^{(k)} \mid \phi(t)\}$ és $S = \{t^{(k)} \mid \psi(t)\}$, azaz R és S már valahogy ki van fejezve sorkalkulussal
 $\implies R \cup S = \{t^{(k)} \mid \phi(t) \vee \psi(t)\}$
többinél ugyanígy ($R(t)$ és $S(t)$ helyett $\phi(t)$ -t és $\psi(t)$ -t írunk). \checkmark

További példák

TERMÉK(GYÁRTÓ, MODELL, TÍPUS)

PC(MODELL, SEBESSÉG, MEMÓRIA, MEREVLEMEZ, CD, ÁR)

LAPTOP(MODELL, SEBESSÉG, MEMÓRIA, MEREVLEMEZ, KÉPERNYŐ, ÁR)

NYOMTATÓ(MODELL, SZÍNES, TÍPUS, ÁR)

A relációk jelentése:

TERMÉK: az adott nevű gyártó gyártja az adott modellszámú és adott típusú (PC, Laptop vagy nyomtató) terméket

PC: modellszám, sebesség megaHz-ben, memória megabájtban, merevlemez gigabájtban, a CD sebessége (pl. 4x), az ár

Laptop: mint PC-nél, plusz a képernyő mérete hüvelykben

Nyomtató: modellszám, színes-e (i/n), típusa (tintasugaras, lézer, mátrix), ára

A modellszámokról feltesszük, hogy egyediek.

- Melyek azok a PC modellek, amelynek sebessége legalább 150?

$$\pi_{\text{MODELL}} (\sigma_{\text{SEBESSÉG} \geq 150} (\text{PC}))$$

- Mely gyártók készítenek legalább egy gigás merevlemezű laptopot?

$$\pi_{\text{GYÁRTÓ}} (\text{TERMÉK} \bowtie \sigma_{\text{MEREVLEMEZ} \geq 1} (\text{LAPTOP}))$$

TERMÉK(GYÁRTÓ, MODELL, TÍPUS)

PC(MODELL, SEBESSÉG, MEMÓRIA, MEREVLEMEZ, CD, ÁR)

LAPTOP(MODELL, SEBESSÉG, MEMÓRIA, MEREVLEMEZ, KÉPERNYŐ, ÁR)

NYOMTATÓ(MODELL, SZÍNES, TÍPUS, ÁR)

- Adjuk meg a B gyártó által gyártott összes termék modellszámát és árát típustól függetlenül!

$$\pi_{\text{MODELL}, \text{ÁR}} (\sigma_{\text{GYÁRTÓ}='B' \wedge \text{TÍPUS}='PC'} (\text{TERMÉK}) \bowtie \text{PC}) \cup$$

$$\pi_{\text{MODELL}, \text{ÁR}} (\sigma_{\text{GYÁRTÓ}='B' \wedge \text{TÍPUS}='LAPTOP'} (\text{TERMÉK}) \bowtie \text{LAPTOP}) \cup$$

$$\pi_{\text{MODELL}, \text{ÁR}} (\sigma_{\text{GYÁRTÓ}='B' \wedge \text{TÍPUS}='NYOMTATÓ'} (\text{TERMÉK}) \bowtie \text{NYOMTATÓ})$$

- Melyek azok a gyártók, akik laptopot gyártanak, de PC-t nem?

$$\text{TERMÉK}_1 = \rho_{\text{TERMÉK}_1} (\pi_{\text{GYÁRTÓ}, \text{TÍPUS}} (\text{TERMÉK}))$$

$$\pi_{\text{GYÁRTÓ}} (\sigma_{\text{TÍPUS}='LAPTOP'} (\text{TERMÉK}_1)) \setminus \pi_{\text{GYÁRTÓ}} (\sigma_{\text{TÍPUS}='PC'} (\text{TERMÉK}_1))$$

TERMÉK(GYÁRTÓ, MODELL, TÍPUS)

PC(MODELL, SEBESSÉG, MEMÓRIA, MEREVLEMEZ, CD, ÁR)

LAPTOP(MODELL, SEBESSÉG, MEMÓRIA, MEREVLEMEZ, KÉPERNYŐ, ÁR)

NYOMTATÓ(MODELL, SZÍNES, TÍPUS, ÁR)

- Melyek azok a gyártók, amelyek gyártanak legalább két, egymástól különböző, legalább 133 Mhz-en működő PC-t vagy Laptopot? (Nincs két azonos modellszám!)

$$R1 = \pi_{\text{MODELL}, \text{SEBESSÉG}} (\text{PC}) \cup \pi_{\text{MODELL}, \text{SEBESSÉG}} (\text{LAPTOP})$$

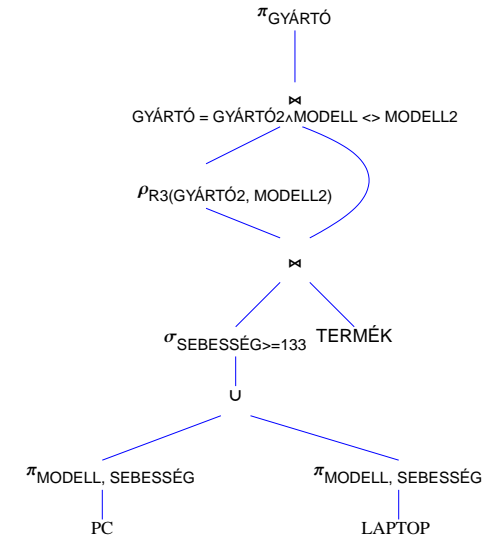
$$R2 = \pi_{\text{GYÁRTÓ}, \text{MODELL}} (\sigma_{\text{SEBESSÉG} \geq 133} (R1) \bowtie \text{TERMÉK})$$

$$R3 = \rho_{R3} (\text{GYÁRTÓ}_2, \text{MODELL}_2) (R2)$$

$$R4 = R2 \bowtie R3$$

$$R5 = \pi_{\text{GYÁRTÓ}} (R4)$$

Megjegyzés: kifejezésfával is meg lehet adni a relációs algebrai kifejezéseket:



További műveletek

Ezek nincsenek benne a relációs algebrában, de fontosak, nem túl műveletigényesek.

aggregátumok: MIN, MAX, AVG, SUM, CNT (szám)

Pl. leggyorsabb gép, átlagár, hányféle printer

eredmény mindig egy szám

aggregátum csoportosítva: Bizonyos feltételek szerinti partíciókban aggregátumok.

Pl. átlagos ár tintasugaras nyomtatók között, egy gyártónak hány terméke van

⇒ eredmény egy reláció pl. (gyártó, szám) párokból.

rekurzív lezárás: (hagyományos adatkezelésben ritka, intelligensebb rendszerekben inkább)

Pl. reláció: ki főnöke kinek ⇒ lezárás: ki felettese kinek

reláció: melyik városból melyikbe van repülő járat ⇒ lezárás: átszállással el lehet-e jutni

Ezt a relációs algebra nem tudja, csak fix mélységre: pl. max 4 átszállás,

(SQL nem igazán támogatja, de azért kicsit igen: SQL3, majd lesz)

Külső illesztés (outer join)

Definíció. R, S relációk $\Rightarrow R \bowtie S$ **bal külső illesztés:** $R \bowtie S$ -hez azokat az R -beli sorokat is hozzáveszük, amihez nem illeszkedik S -beli. Hiányzó helyekre NULL kerül.

Pl. SZEMÉLY(NÉV, KÓD), CÍM(KÓD, CÍM)

$SZEMÉLY \bowtie CÍM \Rightarrow$ akinek nincs címe nem lesz rajta

$SZEMÉLY \bowtie CÍM \Rightarrow$ kiderül, kinek nincs meg a címe

SQL-ben van, relációs algebrával elvileg nem fejezhető ki (NULL miatt), de elkerülhető.

Ha a relációs algebrát úgy definiáljuk, hogy kiindulhatunk konstans relációból is, akkor:

$(R \bowtie S) \cup (R \setminus (R \bowtie S)) \times \{(NULL, \dots, NULL)\}$

Van **jobb külső illesztés** is: $R \bowtie S$

Teljes külső illesztés: $R \bowtie S := (R \bowtie S) \cup (R \bowtie S)$

Példa:

R	A	B	C	S	D	C	$R \bowtie S$	A	B	C	D
a	b	2		a	2		a	b	2	a	
a	c	3		b	3		a	b	2	x	
b	a	4		x	2		a	c	3	b	
				y	1		b	a	4	NULL	

A NULL érték

Lehet, hogy vannak kitöltetlen mezők, ezt meg akarjuk engedni. Pl. ha csak a termelő anyjának neve hiányzik, attól még a termék ára kell. ⇒ **NULL érték**

2 alapvető értelmezés (majd SQL-nél lesz, hogy hogyan kell megmondani, hogy melyik van éppen, illetve, hogy lehet-e egyáltalán NULL valahol):

- \nexists

- \exists , de nem ismerjük.

Attól függően, hogy hogyan értelmezzük a NULL-t:

Mi legyen egy ilyen kérdéssel?:

Pl. $\pi_{CÍM=BP}$ (TERMELŐ)

Ilyenkor belevegyük-e ha a cím NULL?

R	A	B	C	S	D	C	$R \bowtie S$	A	B	C	D
a	b	2		a	2		a	b	2	a	
a	c	3		b	3		a	b	2	x	
b	a	4		x	2		a	c	3	b	
				y	1		b	a	4	NULL	
							NULL	NULL	1	y	

Külső unió

Részben kompatibilis relációk egyesítésére:

DIÁK(NÉV, TÉMAVEZ, TSZK)

TANÁR(NÉV, TSZK, BEOSZT)

	DIÁK \cup_k TANÁR			
	NÉV	TSZK	TÉMAVEZ	BEOSZT
diák				NULL
tanár			NULL	

ODL séma átírása relációsémává

Legegyszerűbb eset \Rightarrow az osztályoknak csak attribútumai vannak, amik atomi típusúak:

```
interface Film (keys (cím, év)){
  attribute cím;
  attribute év;
  attribute hossz;
  attribute szalagFajta;
};
 $\Rightarrow$  Film(cím, év, hossz, szalagFajta)
```

A kulcs az ODL-es kulcs lesz, ha egy van csak belőle; ezt aláhúzással jelöljük. Ha több kulcs van: egyiket jelöljük, a többit írásban (de persze azok is kulcsok a relációsémában is).

Összetett típusú attribútumok: pl. rekordszerkezet OK \Rightarrow

```
interface Színész (key név){
  attribute String név;
  attribute Struct Cím
  { string város, string utca } lakcím;
};
 $\Rightarrow$  Színész(név, város, utca)
```

Kulcs: ugyanúgy, mint a nem összetett attribútumnál

Multihalmazos szemantika

A relációs algebrában ugyan minden reláció halmaz, ezért nincsenek többszörös sorok, de pl. SQL-nél **lesznek**. A multihalmazokkal kicsit máshogy vannak a halmazműveletek:

Ha a t sor $m_R(t)$ példányban van meg R -ben és $m_S(t)$ példányban van meg S -ben, akkor

- $m_{(R \cup S)}(t) := m_R(t) + m_S(t)$ példányban lesz meg R és S uniójában
- $m_{(R \cap S)}(t) := \min\{m_R(t), m_S(t)\}$ példányban lesz meg R és S metszetében
- $m_{(R \setminus S)}(t) := \max\{m_R(t) - m_S(t), 0\}$ példányban lesz meg $R \setminus S$ -ben

- Kiválasztásnál, vetítésnél nincs változás.

De pl. halmaz csak rosszul oldható meg: halmaz minden eleméhez új sor.

```
interface Színész (key név) {
  attribute String név;
  attribute Set <
  Struct Cím{ string város, string utca }
  > lakcím;
};
 $\Rightarrow$  Színész(név, város, utca)
```

név	város	utca
Gálvölgyi J.	Budapest	Nyereg u. 2.
Gálvölgyi J.	Budapest	Kantár u. 3.
⋮		

Kulcs: elromlik az ODL-es kulcs, lehet, hogy ami ott kulcs volt, itt már nem lesz az \Rightarrow baj

Kapcsolatok átírása

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

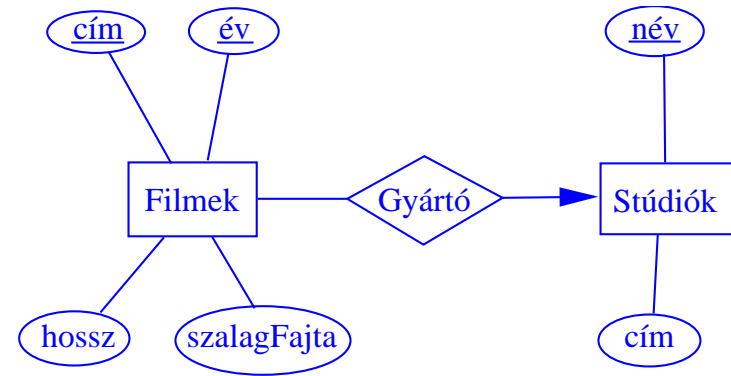
```
interface Film (keys (cím, év)){
  attribute cím;
  attribute év;
  attribute hossz;
  attribute szalagFajta;
  relationship Stúdió gyártó
  inverse Stúdió::gyárt;};
```

⇒ Film(cím, év, 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 ⇒ katasztrófa



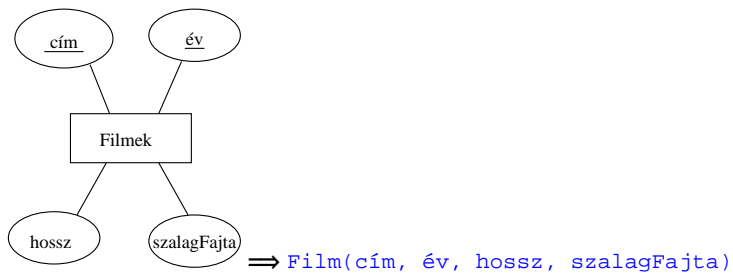
⇒

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

E/K-ból jobban lehet relációt csinálni.

E/K modell átírása

Egyedhalmaz attribútumokkal:



A reláció kulcsa = az egyedhalmaz kulcsa

Egyértékű és többértékű kapcsolat:

⇒ Külön reláció, attribútumai: a kapcsolatban résztvevő egyedhalmaz kulcsainak uniója + kapcsolat attribútumai (esetleg átnevezés)

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

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 be vesszük 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

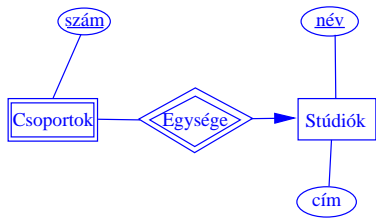
Film(cím, év, hossz, szalagFajta, **stúdióNév**)

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) \Rightarrow Egység(szám, név) hiszen ugyanaz kétszer.
 \Rightarrow 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 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 kulcsa a főosztály kulcsa lesz, így a kapcsolatba is ezt viszi magával az alosztály. Az "isa" kapcsolathoz nem rendelünk relációt.

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.)

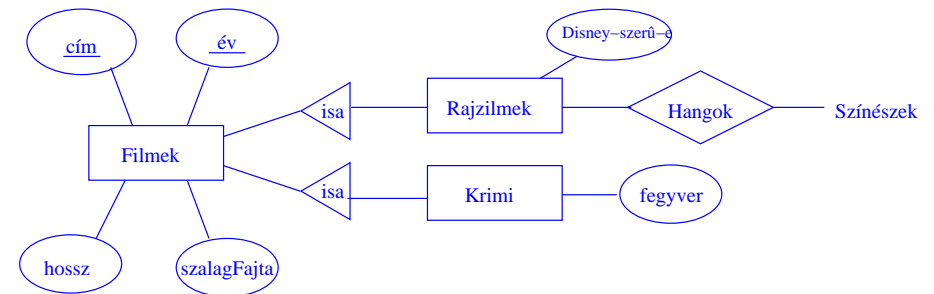
Minden alosztályhoz egy reláció, minden attribútumával és kapcsolatával (öröklöttekkel is).

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 öröklí 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



\Rightarrow
 Film(cím, év, hossz, szalagFajta)
 Rajzfilm(cím, év, Disney-szerű-e)
 Krimi(cím, év, fegyver)
 Hangok(cím, év, Színésznév)

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

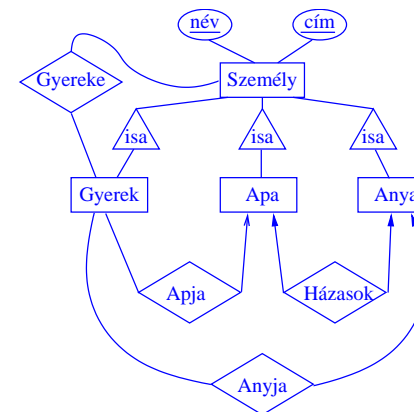
Film(cím, év, hossz, szalagFajta, stúdióNév, színésznév, fegyver)

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

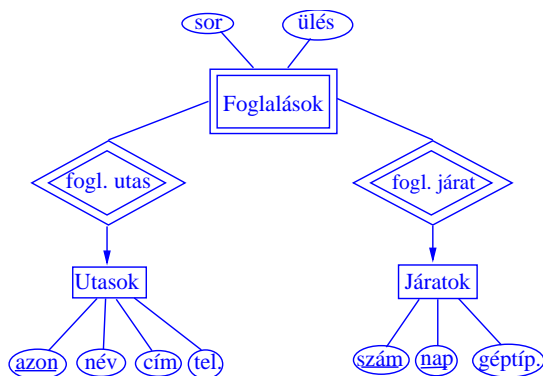
Példák



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, szülónév, szülőcím)

Példák

Alakítsuk relációssá:



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

Megjegyzés: ha a Foglalások nem gyenge egyedhalmaz lett volna, hanem kapcsolat a két egyedhalmaz között, akkor is ugyanez a séma relációs jött volna ki.