

## Adatbázisok, 1. gyakorlat: Az egyed-kapcsolat modell

### Feladatok

- 1) Filmeket, azok rendezőit, színészeit, forgatókönyvíróit kívánjuk tárolni. A filmekről tároljuk a hosszukat, a megjelenésük dátumát, a megtekintésükhöz ajánlott minimális életkort és a címüket. A cím egyértelműen azonosítja a filmet. Rendezőkről, színészekről és forgatókönyvírókról tároljuk a nevüket, címüket, személyi és telefonszámukat. A személyi szám egyértelműen azonosít minden személyt.
  - a. Tervezze meg a feladatnak megfelelő EK-modellt!
  - b. Az ER-modellben tárolni szeretnénk, hogy az egyes színészek mely filmekben mely szereplőt játsszák (tfh. egy színész egy filmben csak egy szerepet játszik). Ennek megfelelően bővítse a diagrammot!
  - c. A filmekről tárolni szeretnénk, hogy krimi esetén ki(k) játsszák az áldozato(ka)t, természetfilmeknél pedig azt, hogy milyen ország(ok)ban forgatták őket.
- 2) Iskolákat, azokon belüli osztályokat valamint tanárokat és diákokat tartunk nyilván. Az iskolákról tároljuk a nevüket, címüket, igazgatóik kilétét. Az osztályokról tároljuk létszámukat, nevüket, osztályfőnökük valamint tanulóik kilétét. A tanárokról tároljuk, hogy melyik osztályokat tanítják, mit tanítanak. Minden emberről továbbá tároljuk a személyes adatokat (név, lakcím, szem szám, születési időpont)
  - a. Tervezze meg az ER-modellt, válassza meg a kulcsokat!
  - b. Az EK-diagrammban tároljuk a munkaviszonyok, hallgatói jogviszonyok kezdetét is.
- 3) Egy lízing-szerződés kapcsán a kereskedő átadja a kocsit az ügyfélnek, a kocsit a bank fizeti ki a kereskedőnek és az ügyfél pedig a banknak törleszt. Milyen módokon lehet ábrázolni EK-diagrammal, melyiknek mi az előnye és a hátránya?

### Gondolkodtató feladatok:

- 1) Az alábbiak közül melyek jelennek meg az EK-modellben és melyek az általa modellezett világban (az EK-modellben milyen formában)?
  - a. egyed
  - b. egyedhalmaz
  - c. kapcsolat
  - d. kapcsolattípus
  - e. attribútum
  - f. attribútumtípus
- 2) Hozzon egy-egy példát arra, hogy a „pénzt” mint attribútumot, illetve mint egyedet tároljuk.
- 3) Lehetséges-e? Mondjunk rá példát! Hogyan jelöljük az EK-diagrammon?
  - a. Két egyedhalmaz között több kapcsolathalmazt veszünk fel.
  - b. Egy-egy kapcsolatból több fut egyedek között.
  - c. Egyedhalmaz önmagával van kapcsolatban.

**ZH-szintű gondolkodtató példa otthonra:** Tankönyv, 217. oldal, 3. feladat (2016-os kiadás szerint, ER-modellezés, 3. gyakorlófeladat, 2006-os kiadásban: 134. oldal, 4. feladat)

22 September 19.

gya81

# Adatb gyaz

adatb@sch.buu.hu

Molnar Balazs

molnar.balazs@db.buu.hu

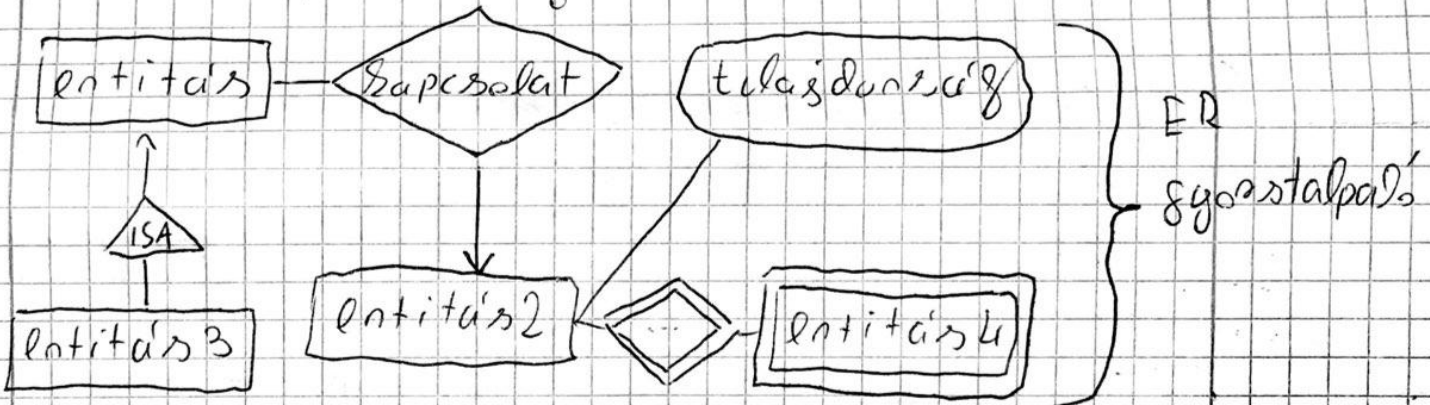
ubi

Modellering, ment a valosag benyolalt.

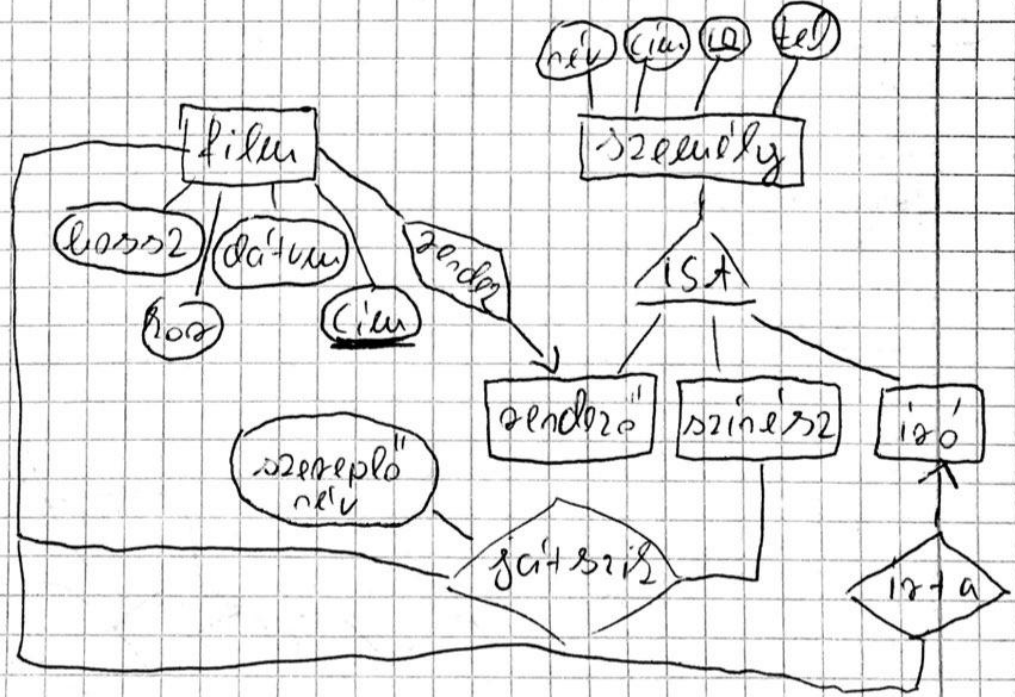
Az ER nem adatmodell, ment muveltetessel nem foglalkozik, de hasznos.

Az alap ER-hoz szingulus bevitmenyok is a determinalo kapcsol

Tipikusan modellezeszo nem a szintot peldanyokkal, hanem altalinositasigban beszélnek.



1.)



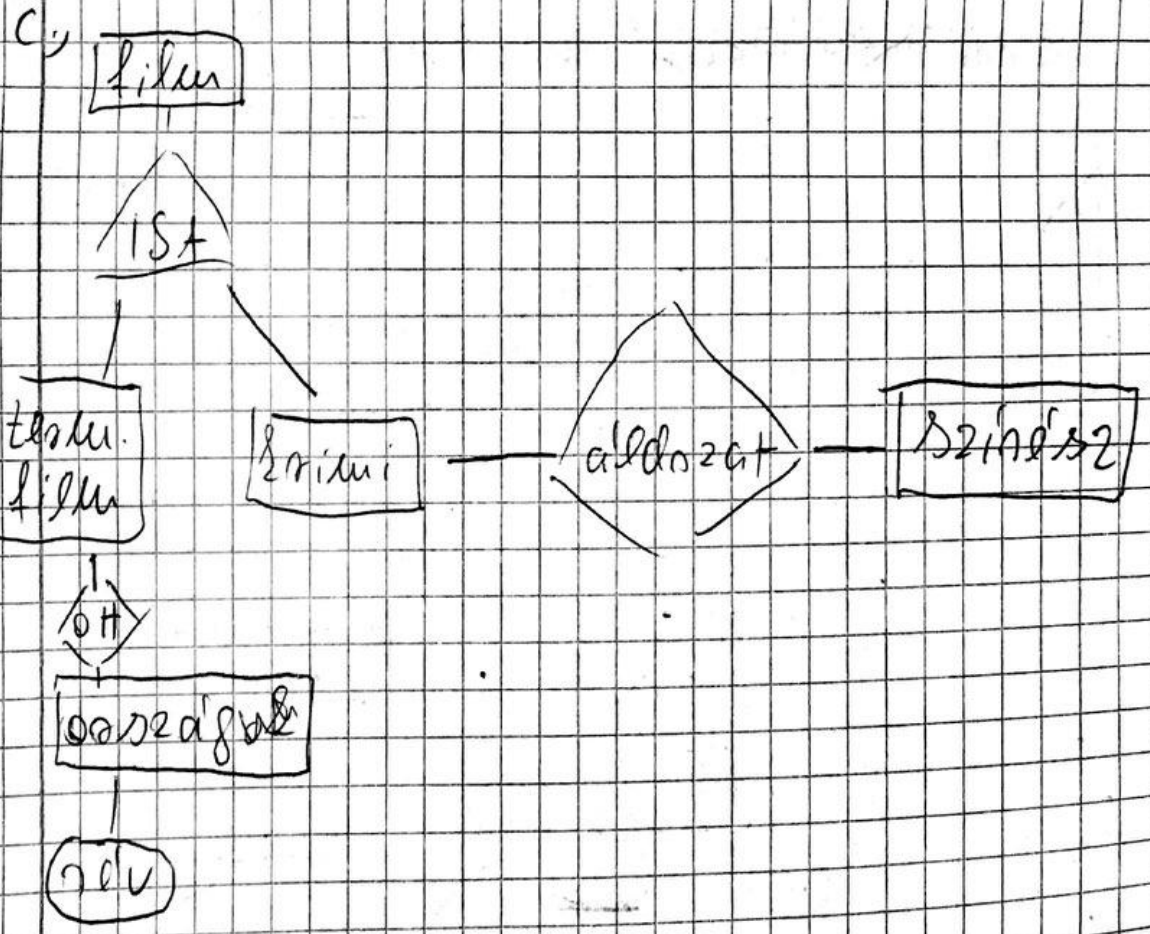
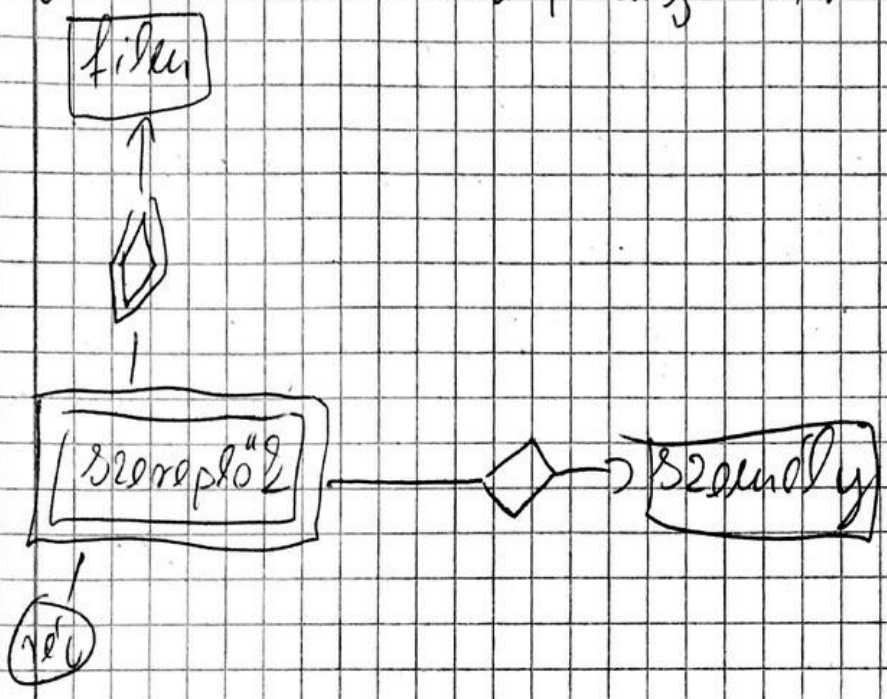
vagy azaz az is lehet, hogy a szereplo benne a kapcsolatoz

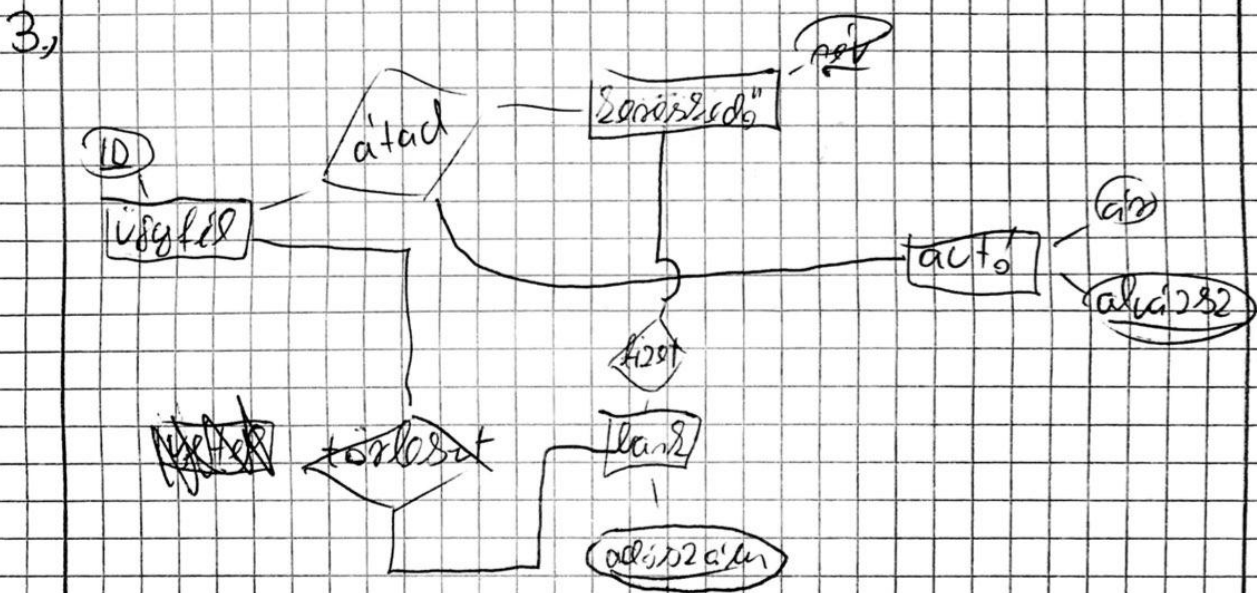
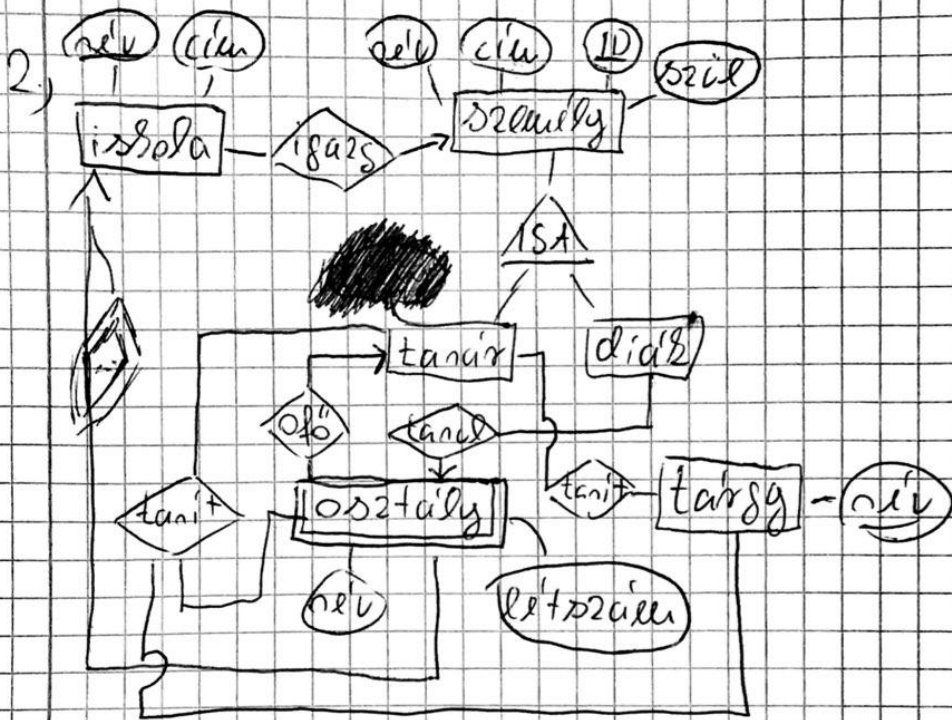
a b-t isy is lehet, hogy van egy + szereplo ent, ami szereplo a film lete.

Filmek (név, dátum, elstbor, cím)

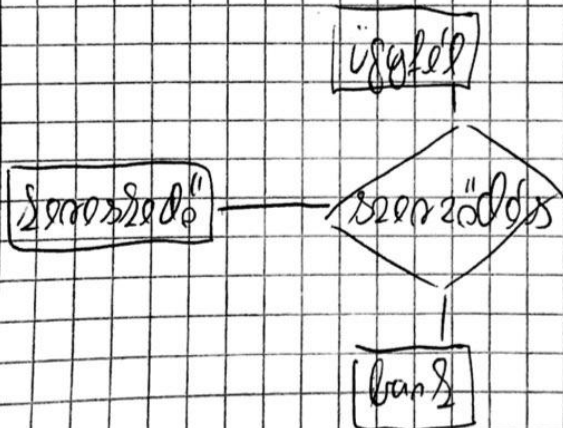
rendező }  
színdísz } (név, cím, id, tel)  
hang.író }

b-kez: (amit a lap alján írtam)





vagy:



1: név	2: egyetem	3: szak	4: kezdés éve	1: egyetem	2: alapítás éve	1: név	2: város
Kiss Aladár	BME	informatikus	2011	SZTE	1581	Kiss Aladár	Budapest
Nagy Béla	BME	gépész	2007	ELTE	1635	Nagy Béla	Győr
Olaf Gergő	CEU	menedzsment	2009	Pázmány	1635	Olaf Gergő	Pozsony
Cenk Mariann	Pázmány	informatikus	2004	BGE	1857	Cenk Mariann	Tata
Takács Márton	BGE	menedzsment	2010	BME	1782	Takács Márton	Kiskőrös
Rác Kata	ELTE	tanár	2011			Rác Kat	Keszthely
						Tóth Ödön	Budapest

A feladatok megoldásához használt példa relációk: **H**(hallgatók), **Á**(államilag támogatott egyetemek) és **S**(zemély) relációk

## Feladatok

- Az alábbi kérdéseket fogalmazzuk meg relációalgebra segítségével!
  - Melyek az államilag nem támogatott egyetemek?
  - Kik azok a hallgatók, akiknek nincs hallgatói jogviszonyuk államilag támogatott egyetemen? (Feltételezzük, hogy egy hallgató egyetlen felsőoktatási intézmény diákja.)
  - Melyek azok a szakok, amelyeket legalább két egyetemen oktatnak?
  - Melyek azok a szakok, amiket csak egy-egy egyetemen oktatnak?
  - Melyik a legrégebben alapított és támogatott egyetem?
- Az előző problémákat fogalmazzuk meg sor- és/vagy oszlopalkulussal segítségével!

## Gyakorló feladatok

Az alábbi kérdéseket fogalmazzuk meg relációalgebra és sor/oszlopalkulussal segítségével!

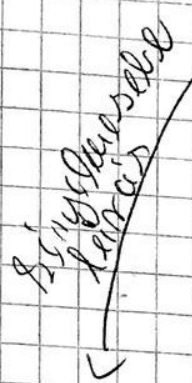
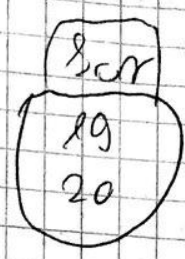
- Mely szakokat tanítanak támogatott és nem támogatott egyetemen is?
- Kik a Budapestről származó informatikushallgatók?
- Mely településekről származnak az államilag támogatott egyetemen hallgatói?
- Kik azok a hallgatók, akiket 2005 előtt vettek fel, vagy államilag nem támogatott egyetemen járnak?

## Gondolkodtató kérdések

Házi feladat: Tankönyv *Relációs lekérdező nyelvek*hez kapcsolódó feladatai közül a 26. feladat.

- A feladatlap elején szereplő relációkból bármely relációalgebrai művelet akárhányszori ismétlésével megkaphatóak-e az alábbi sorok? Ha igen, adjuk meg a műveleteket, ha nem, akkor bizonyítsuk be miért nem!
  - (Kiss Aladár, Budapest, Olaf Gergő, Pozsony, CEU)
  - (Kiss Aladár, Budapest, SOTE)
- Lehetséges-e ugyanezeket az eredményeket megkapni sor- és/vagy oszlopalkulussal?
- Készíts relációalgebrai kifejezést egy halmaz legkisebb, illetve második legkisebb elemének kiválasztására! Mi az ennek megfelelő sor- és oszlopalkulussal kifejezés?
- Mondjunk minél kacifántosabb helybenhagyó műveleteket!
- Mikor kényelmesebb a sor és mikor az oszlopalkulussal?
- Egy Apa-Fia relációból keresd ki azokat, akik nem nagyapák!
- Az egyes relációalgebrai kifejezésekről állapítsuk meg, hogy az eredményül kapott reláció kisebb vagy nagyobb, mint az eredeti, illetve mikor lehet azonos méretű (azaz az eredmény reláció sor és oszlopszáma, hogyan viszonyul az eredeti reláció(k) azonos paramétereikhez)?
- Mely relációalgebrai műveletek invertálhatók, és milyen módon az eredeti reláció(k) felhasználása nélkül?
- Soroljuk fel a sor- és oszlopalkulussal közötti átirás főbb lépéseit!

Halmazok:



neptun x sor = (BATMAN, 19);  
 (BATMAN, 20);  
 (SOSY61, 19);  
 (SOSY61, 20)

neptun	sor
BATMAN	19
SOSY61	20

1.)

a.)  $\underline{NT} = \tilde{\Pi}_{\text{egyetem}}(H) \setminus \tilde{\Pi}_{\text{egyetem}}(A)$

b.)  $\tilde{\Pi}_{(n \times v)}(H) \setminus (\tilde{\Pi}_{(n \times v)} \delta_{(H.\text{egyetem} = A.\text{egyetem})}(H \times A'))$   
 vagy egyszerűbben

$\tilde{\Pi}_{(n \times v)}(H) \setminus \tilde{\Pi}_{(n \times v)}(H \times A')$

vagy akár

$\tilde{\Pi}_{(n \times v)}(H \times NT)$

c.)  $\underline{C} = \tilde{\Pi}_{(s2a2)}(\delta_{(H1.\text{egyetem} \neq H2.\text{egyetem} \wedge H1.s2a2 = H2.s2a2)}(H1 \times H2))$   
 vagy egyszerűbben

$\parallel H = H1 = H2$

$\tilde{\Pi}_{(s2a2)}(H1 \quad | \quad H2)$

$H1.\text{egy} \neq H2.\text{egy}$   
 $\uparrow$   
 $H1.s2a2 = H2.s2a2$

note!  
 a pi kiszeli a2  
 ismeretlenek

$$d.) \tilde{\pi}_{\langle \text{szabvány} \rangle} (H) \setminus C$$

$$e.) A^1 = A^2 = A^3 = A'$$

$\tilde{\pi}_{(A^1 \text{-esgy})} (A^1 \setminus \tilde{\pi}_{(A^2 \text{-esgy}, A^2 \text{-alap})} \{ (A^2 \text{-alap}) \rightarrow A^3 \text{-alap} \} (A^2 \times A^3))$   
 vagy egyszerűen

$$\tilde{\pi}_{(A^1 \text{-esgy})} (A^1 \setminus \tilde{\pi}_{(A^2 \text{-esgy}, A^2 \text{-alap})} (A^2 \times A^3))$$

$A^2 \text{-alap} \rightarrow A^3 \text{-alap}$

$\exists, \forall$  kvantorok

a sor- és oszlopkezelés kihasználásával legalább akkora, mint a relációs algebra

2.) Ugyanez sor- és / vagy oszlopkezeléssel

$$a.) \left\{ \begin{array}{l} \exists s^{(1)} \mid (\exists u^{(4)}) H^{(u)} (u^{(4)}) \wedge \exists [1] = u^{(1)} [2] \wedge \\ \wedge (\forall a^{(2)}) \vee (a^{(2)}) \wedge a^{(2)} [1] \neq s^{(1)} [1] \end{array} \right\}$$

ide jön az eredmény

létezik olyan  $u$ -es sorvektor, ami nem  $H$ -beli vagy

$$\left\{ \begin{array}{l} \exists s^{(1)} \mid (\exists u^{(4)}) H^{(u)} (u^{(4)}) \wedge \exists [1] = u^{(1)} [2] \wedge \\ \wedge (\exists a^{(2)}) \wedge a^{(2)} [1] = s^{(1)} [1] \end{array} \right\}$$

$$b.) \left\{ t^{(1)} \mid (\exists u^{(4)}) H^{(4)}(u^{(4)}) \wedge t^{(1)} [1] = u^{(4)} [1] \wedge \right. \\ \left. \wedge \neg ((\exists a^{(2)}) \wedge a^{(2)} [2] = a^{(2)} [1]) \right\}$$

c.) ez most országhalgelesselel

$$\left\{ t \mid (\exists x, y, z, v, w, q) H(x, y, t, z) \wedge \right. \\ \left. \wedge H(v, w, t, q) \wedge w \neq y \right\}$$

iH az attribútumok sorrendje számít, nyilván  $w \neq y$  az egyetlen.

vszámor országhalgelesselel

$$\left\{ t^{(1)} \mid (\exists u^{(4)}, \delta^{(4)}) H^{(4)}(u^{(4)}) \wedge H^{(4)}(\delta^{(4)}) \wedge \right. \\ \left. \wedge u^{(4)} [3] = \delta^{(4)} [3] \wedge u^{(4)} [3] = t^{(1)} [1] \wedge \right. \\ \left. \wedge u^{(4)} [2] \neq \delta^{(4)} [2] \right\}$$

valamikor az egyik kifejezésből, mint a másiké de persze bármelyikből is lehet fejteni a másikat

d.) ez is országhalgelesselel

$$\left\{ d \mid (\exists x, y, z) H(x, y, d, z) \wedge (\forall v, w, q, t) \neg H(v, w, q, t) \vee \right. \\ \left. \vee d \neq q \vee y = w \right\}$$

iH  $d$  az a számsz



ε) Σορβαλλοσυστάση:

$$\left\{ t^{(1)} \mid (\exists t^{(2)}) A^{(2)}(t^{(2)}) \wedge t^{(2)} [1] = \lambda^{(1)} [1] \wedge ((\forall x^{(2)}) \top A^{(2)}(x^{(2)}) \vee \right.$$
$$\left. \vee x^{(2)} [1] = \lambda^{(1)} [1] \vee x^{(2)} [2] > t^{(2)} [2]) \right\}$$

---

συμβολισμός/2,

Κιζ α Βυδαπεστσολ σζα'ρμαζο' ινφορματίζουσ μαθηζατοί? ζολα'ϊοαλζελκίναλ:

ii) (H. α'ελ)  $\tilde{\sigma}_{H.2} = \text{'informatics'} \wedge S.2 = \text{'Budapest'} \wedge S.1 = H.1$  (S x H)  
οσζλο,ρζαλλζοσυστάση:

$$\left\{ w \mid (\exists y, z, v) S(w, t) \wedge t = \text{'Budapest'} \wedge H(w, y, z, v) \wedge \right.$$
$$\left. \wedge z = \text{'informatics'} \right\}$$

ιαζγ ρο'νιδελκίναλ:

$$\left\{ w \mid (\exists y, v) S(w, \text{'Budapest'}) \wedge H(w, y, \text{'informatics'}, v) \right\}$$

**Keretmese** Egy szerszámkölcsonzó adatbázisában a következő adatokat tároljuk: milyen termékkódú, felhasználási célú és méretű eszközből mekkora fajlagos kölcsönzési díjért milyen igazolványszámú ügyfélnél hány darab van kölcsönben, összesen mekkora készletből, amelynek mekkora része található még szabad, kikölcsönözhető állapotban a kölcsönző melyik polcán. Tudjuk továbbá, hogy a termékkód egy eszköztípust (adott gyártó adott terméke) egyedileg azonosít; az azonos felhasználási célú (kalapács, hegesztő stb.) és méretű (1 kilogrammos, 400 wattos stb.) eszközöket típustól függetlenül azonos díjért adják bérebe és mindig egyazon polcon tárolják; egy polcon pedig csak egyféle célú eszközök sorakoznak.

### Kezdeti, univerzális séma

- $R(Igazolványszám, Termékkód, Kölcsönzött\ db, Összes\ db, Szabad\ db, Cél, Méret, Polc, Díj) = R(ITKÖSCMPD)$
- $F = \{IT \rightarrow K, T \rightarrow ÖSCM, CM \rightarrow PD, P \rightarrow C\}$

### Feladatok

1. Van-e a tárolásban a jelen állapotban redundancia?
2. A redundanciacsökkentés érdekében ellenőrizzük, hogy milyen legmagasabb normálformában van a séma!
3. Találjunk veszteségmentes sémafelbontást minél kevesebb számú, legalább 2NF részsémába!
4. A megalkotott adatbázisséma 3NF-ben van-e? Ha nem, mutassunk példát redundanciára, és a redundancia további csökkentése érdekében találjunk veszteségmentes sémafelbontást minél kevesebb számú, legalább 3NF részsémába!
5. A megalkotott adatbázisséma BCNF-ben van-e? Ha nem, mutassunk példát redundanciára, és a redundancia további csökkentése érdekében találjunk veszteségmentes sémafelbontást minél kevesebb számú, legalább BCNF részsémába!
6. Lehet-e funkcionális függés alapú redundancia az így létrejött felbontás alapján elkészített adatbázisban? Függőségőrző-e a sémafelbontás? Ha nem, ennek milyen hátrányos következményei vannak?
7. Az eredeti relációs sémát a tanult módszerrel bontsuk fel veszteségmentes és függőségőrző módon legalább 3NF részsémákba! Törekedjünk arra, hogy minél kevesebb számú részséma keletkezzen!
8. Ellenőrizzük a táblázatos módszer segítségével, hogy az előbb elkészített sémafelbontás valóban veszteségmentes-e!

### Gondolkodtató kérdések

1. Ugyanezt a szerszámkölcsonzós példát alapul véve, van-e olyan (nyilván nem funkcionális függőség alapú) redundancia az adatbázisban, amit egyik felbontás se küszöbölt ki (tehát még a BCNF sem)? Hogyan küszöbölhető ki? Megéri?
2. Adott egy séma. Milyen függőség típusokra lehet a pillanatnyi állapotból következtetni és milyen feltételekkel?
3. Hányféle „kulcs” fogalom került eddig elő a tantárgyban? Mik a különbségek?
4. Hányféle „lezárt” fogalommal találkoztunk? Különböznek-e egymástól? Mi a kapcsolat köztük? Hogyan számíthatóak?
5. Egy sémájával adott relációs adatbázis hogy lesz ténylegesen eltárolva? Találjunk kapcsolatot a fizikai szervezés témakörében megismert fogalmakkal!
6. Az adatbázis sémájának felbontása hogyan érinti az adatmanipulációt? Találjunk kapcsolatot a lekérdezések témakörében megismert fogalmakkal! Írjunk fel néhány több táblát érintő lekérdezést relációs algebra vagy sor/oszlopkalkulus segítségével!
7. Hogyan kell EK modellekből relációs adatbázissémát készíteni?
8. Egy EK modellből milyen funkcionális függőségek olvashatók le? Ha ezeken kívül más függőség nem adott, akkor mit mondhatunk az ilyen módon származtatott adatbázisséma normálformájáról?
9. Találjunk ki olyan (értelmes) adatbázissémát, amelyik nem 1NF!
10. Adatbázisunk összes relációjának sémája mindössze két számértékű és egy karakterlánc attribútumból áll. Biztosan állíthatjuk-e, hogy az adatbázis legalább 1NF?
11. A BCNF-fel ellentétben a csupán 3NF séma okozhat funkcionális függés alapú redundanciát. Tudunk-e érveket felhozni amellett, hogy ezen redundancia jelentősége és hatása várhatóan kisebb a csak alacsonyabb normálformák esetén felbukkanó redundanciáknál?
12. Milyen hatással lehet a adatbázisséma különböző tanult tulajdonságaira, ha egy meglévő sémafelbontást egy újabb részsémával bővítünk? Továbbá milyen a hatása annak, ha egyik részsémáját sémafelbontással tovább bontjuk?
13. Adott ugyanannak a sémának két különböző felbontása. A két új adatbázisséma normálformában megegyezik. Milyen szempontok alapján érdemes közülük választani?
14. A relációs sématervezés témakörben kimondott tételek közül melyiknek van a legtöbb közvetlen gyakorlati jelentősége?

**Házi feladat** Példatár 42., 50., 54., 56., 60., 61. feladat.

### Normálforma definíciók

- 2NF: 1NF és  $\forall K$  kulcs  $\forall A \in R$  másodlagos:  $A$  teljesen függ  $K$ -től.
- 3NF/i.: 1NF és  $\forall K$  kulcs  $\forall A \in R$  másodlagos:  $A$  nem függ tranzitívan  $K$ -től.
- BCNF/i.: 1NF és  $\forall K$  kulcs  $\forall A \in R$ :  $A$  nem függ tranzitívan  $K$ -től.
- 3NF/ii.: 1NF és  $\forall X \subseteq R$   $\forall A \in R, X \rightarrow A$  nemtriviális:  $X$  szuperkulcs  $\vee A$  elsődleges.
- BCNF/ii.: 1NF és  $\forall X \subseteq R$   $\forall A \in R, X \rightarrow A$  nemtriviális:  $X$  szuperkulcs.

Igazolványszám	Termékkód	Kölcsönzött db	Összes db	Szabad db	Cél	Méret	Polc	Díj
123456AB	1	4	20	14	fűrógép	800 W	A	2000
123456AB	2	3	15	12	kalapács	0,5 kg	B	1000
314159QF	3	1	5	4	kalapács	1 kg	B	1500
314159QF	4	2	5	3	kalapács	0,5 kg	B	1000
314159QF	1	2	20	14	fűrógép	800 W	A	2000

1. táblázat. Az  $R$  1NF sémára illeszkedő példa reláció

Igazolványszám	Termékkód	Kölcsönzött db
123456AB	1	4
123456AB	2	3
314159QF	3	1
314159QF	4	2
314159QF	1	2

2. táblázat. Az  $R_1(ITK)$  BCNF sémára illeszkedő példa reláció

Termékkód	Összes db	Szabad db	Cél	Méret	Polc	Díj
1	20	14	fűrógép	800 W	A	2000
2	15	12	kalapács	0,5 kg	B	1000
3	5	4	kalapács	1 kg	B	1500
4	5	3	kalapács	0,5 kg	B	1000

3. táblázat. Az  $R_2(TÖSCMPD)$  2NF sémára illeszkedő példa reláció

Cél	Méret	Polc	Díj
fűrógép	800 W	A	2000
kalapács	0,5 kg	B	1000
kalapács	1 kg	B	1500

4. táblázat. Az  $R_3(CMPD)$  3NF sémára illeszkedő példa reláció

Termékkód	Összes db	Szabad db	Cél	Méret
1	20	14	fűrógép	800 W
2	15	12	kalapács	0,5 kg
3	5	4	kalapács	1 kg
4	5	3	kalapács	0,5 kg

5. táblázat. Az  $R_4(TÖSCM)$  BCNF sémára illeszkedő példa reláció

Polc	Cél
A	fűrógép
B	kalapács

6. táblázat. Az  $R_5(PC)$  BCNF sémára illeszkedő példa reláció

Méret	Polc	Díj
800 W	A	2000
0,5 kg	B	1000
1 kg	B	1500

7. táblázat. Az  $R_6(MPD)$  BCNF sémára illeszkedő példa reláció

október 17.

89 a 83

de. lwe. lwd - calculator

applet funkcionális függőségek, ellenőrizni

islatás normalizálással ellenőrizni a redundanciát.  
elkerülése érdekében:

• funkcionális függés:  $t[x] = t'[x] \rightarrow t[y] = t'[y]$

eset, utólagosok, az új  
névűk sirdom:  $t, t'$ :  $t, t'$

• mindenféle normalizálás, BCNF-re közeledés

• teljes:  $X \rightarrow R$ ,  $\exists X' \subset X$   $X' \rightarrow R$

• szuperjelcs:  $X \rightarrow R$

• arisztótelos-axiómákkal tudni kell, de most nem kell

• attribútumok lezárása:

$$X^+(F) = \{A \mid A \in R, F \models X \rightarrow A\}$$

• függőség lezárása

• vezetékes lezárás:  $\forall \alpha(R) \cap \pi_{R_1}(\alpha) \cap \dots \cap \pi_{R_n}(\alpha) = \alpha$

ha a reláció teljesüljön az  
első reláció

$\{R_1, R_2\}$  vezetékes lezárás

$(R_1 \cap R_2) \dots$  jogszerűen lenne van

1.) igen, lehet van vele

2.) hogyan érdekeltes ezt?

- ellenőrizni, hogy IMF-e (néhány más lesz, hogy nem)

- nézni meg, hogy BCF-e

- nézni meg, hogy ZNF-e

- nézni meg, hogy ZNF-e

és a maradék is jó, mint mindig.

nézzük meg, hogy a függőség valóban szerepel-e

$$IT^+(F) = IT \underbrace{K}_{IT \rightarrow K} \underbrace{ÖSKM}_{I \rightarrow ÖSKM} \underbrace{PD}_{CM \rightarrow PD} \quad \checkmark \quad \text{szuper bűvés}$$

$$I^+(F) = I \underbrace{ÖSKM}_{I \rightarrow ÖSKM} \underbrace{PD}_{CM \rightarrow PD} \quad \times \quad \text{nem szuper bűvés, elrontja a BCF-t}$$

$$CM^+(F) = CM \quad \times$$

$$P^+(F) = P \quad \times$$

vagy BCF, ZNF-e?

függőség bal oldala szuper bűvés vagy jobb oldala érdekes utra.

alul szuper bűvés: > ez az utra bűvés

bal	minimális	jobb	egyik oldal
IT	CM	ÖSKD	—

IT megkapta az egész sémát a fenti leírást kiindulva

bűvés: IT

érdekes: IT

maradékos: (MPKÖSD)  $\rightarrow$  azaz bűvés csak a bal oldalt látja meg, és elrontja a ZNF-t

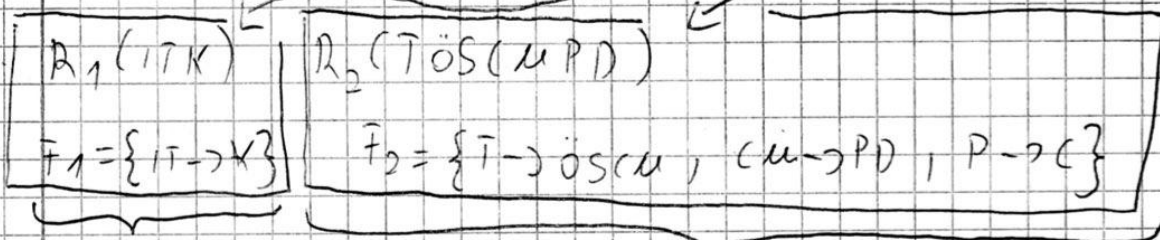
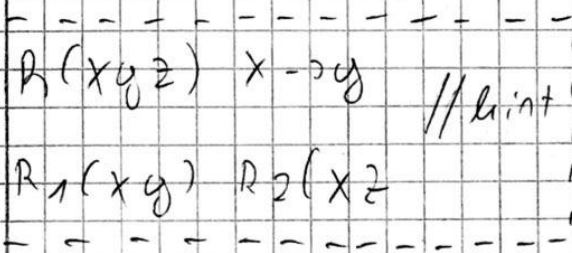
a séma nagyobb nem 3NF, 2NF legalább?

$$1^+(F) = 1 \checkmark$$

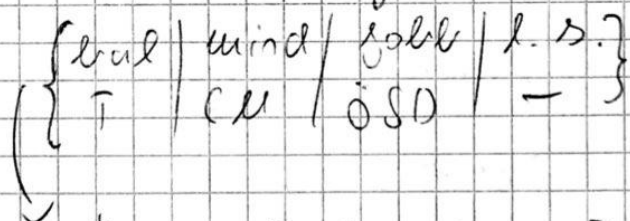
$1^+(F) = T \overset{\circ}{O} S C M P D \quad X \rightarrow$  ez érinti a 2NF-t,  
a séma 1NF  $\leftarrow$  így

2.) veszteségmentes lelkontás legalább 2NF sémára  
indulunk az univerzális  $R(T \overset{\circ}{O} S C M P D)$  1NF sémából

mivel T rontóHa el a dolgozat, ebből kezdés  
indulni.



ez BCNF,  
szóval létezik



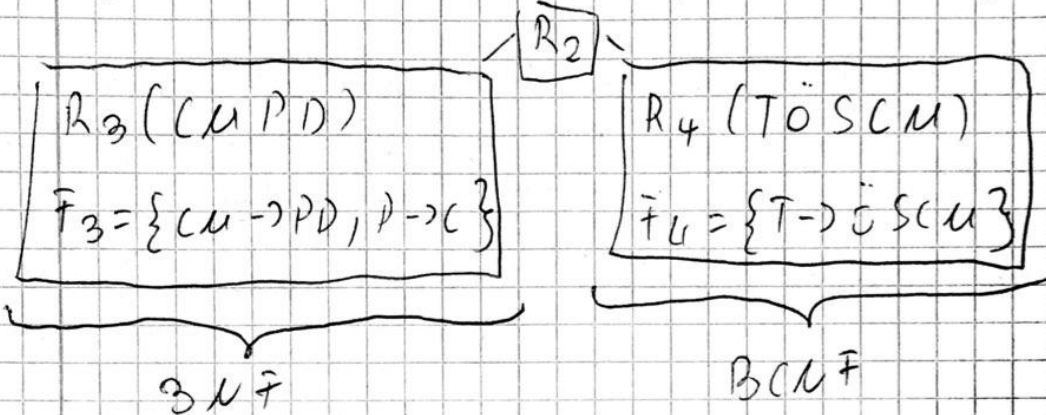
$$1^+(F_2) = T \overset{\circ}{O} S C M P D \Rightarrow T \text{ kulcs}$$

- 3NF-t érinti csomó minden  $(C M \rightarrow P D, P \rightarrow C)$
- 2NF egyértelmű, hiszen csak egy csomó kulcs van.

ez ilyen mechanikus módszer, ami a működés  
pontosan nem tudja nézni a lelkontást.

4.) előbb megmutatjuk hogy nem ZNF

most el lehet indolni  $CM \rightarrow PD$  vagy  $P \rightarrow C$  mentén. erre lesz egy algoritmus, de most egyszerűen, én  $CM \rightarrow PD$ -vel dolgozom.



C	M	P	D
M	C	P	D

$\rightarrow MC^+(F_3) = MCPD \checkmark$

$MP^+(F_3) = MPCD \checkmark$

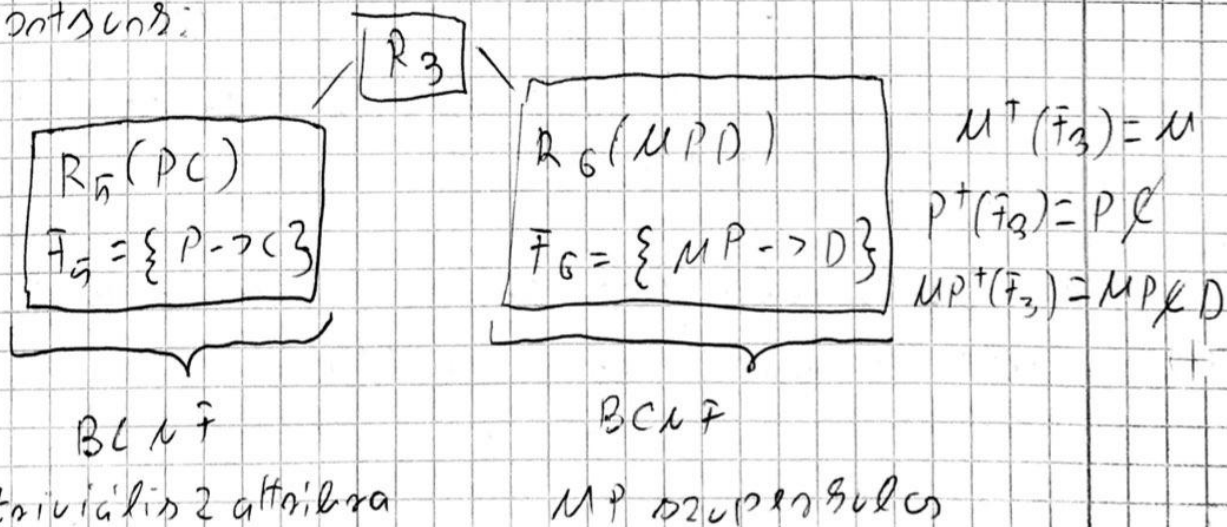
ZNF azért teljesül, viszont P nem szuperkulcs, így BCNF nem.

5.) BCNF-e a reláció? előbb beláthatjuk hogy nem.

$PM \rightarrow P, P \rightarrow PM, P \rightarrow C$

$C \ \& \ P \Rightarrow PM \rightarrow C$

$P \rightarrow C$  miatt ez a BCNF-e, így a mentén vizsgálunk:



6.) newiwert az a B, CUF legyél.

$$G = \{ IT \rightarrow K, \bar{T} \rightarrow \text{öSCM}, P \rightarrow C, MP \rightarrow D \}$$

$$\bar{F} = \{ IT \rightarrow K, T \rightarrow \text{öSCM}, CM \rightarrow PD, D \rightarrow C \}$$

$$G = \bar{F} \Leftrightarrow G^+ = \bar{F}^+$$

$$IT^+(G) = ITK \in K \checkmark$$

$$MP^+(G) = MP \in D \checkmark$$

$$CM^+(G) = CM \& PD \times \text{nem független}$$

7.)

$$F = \{ IT \rightarrow K, \bar{T} \rightarrow \text{öSCM}, CM \rightarrow PD, P \rightarrow C \}$$

$$\textcircled{1} F' = \{ IT \rightarrow K, \bar{T} \rightarrow \text{ö}, T \rightarrow S, \bar{T} \rightarrow C, \bar{T} \rightarrow M, CM \rightarrow PD, CM \rightarrow D, P \rightarrow C \}$$

meg kell nézni, hogy a függetlenek bal oldalainál elmagyarázható-e valahányszor.

②

MAH  $IT \rightarrow K$  köl nem tudunk elmagyarázni

$CM \rightarrow PD, CM \rightarrow D$  köl nem

van-e olyan független, ami elmagyarázható?

$$\textcircled{3} IT^+K = IT \text{öSCMPD} \& K$$

$$T^+ = TSCMPD \& \text{ö}$$

$$T^+ = T \text{öSCMPD} \& S$$

$$T^+ = T \text{öSM} \& C$$

$$T^+ = T \text{öSC} \& M$$

az összes többlet, nem lehet elmagyarázni semmit, megvan a minimális fedés

③



$a$  udg  $\{ \underline{ITK}, T\ddot{o}, TS, T(C)UM, CMP, CPD, PC \}$   
 összevonható, CMPD  
 a régi BCUF  
 R 4 lesz belőle

ez egy olyan z.h.f., ami függőségen 2<sup>n</sup> keresztengmentes  
 ellenőrzés található:

$\{ ITK, CMPD, T\ddot{o}SCM \}$

	I	T	K	ö	S	C	M	P	D
ITK	a	a	a	<del>a</del> <sup>a</sup>	<del>a</del> <sup>a</sup>	<del>a</del> <sup>a</sup>	<del>a</del> <sup>a</sup>	b <sub>1</sub>	b <sub>1</sub>
CMPD	b <sub>2</sub>	b <sub>2</sub>	b <sub>2</sub>	b <sub>2</sub>	b <sub>2</sub>	a	a	a	a
T <sup>o</sup> SCM	b <sub>3</sub>	a	b <sub>3</sub>	a	a	a	a	b <sub>3</sub>	b <sub>3</sub>

ez valahogy nem jó!

$\bar{T} = \{ IT \rightarrow K, MP \rightarrow D, T \rightarrow \ddot{o}SCM \} \cup \{ CM \rightarrow PD \}$

12 szabvány, hogy  
 töltsd nem tudsz  
 radírni.

**1. feladat**

Egy 1000 rekordból álló állományt ritka index szervezéssel tárolunk. A rekordhossz 850 bájt, egy blokk kapacitása (a fejrész nem számítva) 4000 bájt. A kulcs 50 bájtos, egy mutatóhoz 18 bájt kell. Egy blokkművelet ideje 5 ms.

- Hány rekord fér el egy blokkban?
- Hány blokkot foglal el az indexstruktúra és mennyit a teljes állomány?
- Melyik szinten, melyik blokkokban és blokkok között követeljük meg a rendezettséget?
- Mennyi ideig tart legfeljebb egy rekord tartalmának kiolvasása, ha feltételezzük, hogy az index struktúra már benne van az operatív tárban?
- Mennyi ideig tart legfeljebb egy rekord tartalmának kiolvasása, ha az index struktúra nem fér el az operatív tárban?

**2. feladat**

Egy 7 vödörrel rendelkező hash tábla leképező függvénye  $h(k) = k \bmod B$ . A következő kulcsú rekordok érkeznek, amelyeket szeretnénk eltárolni: 56, 91, 27, 19, 36, 52, 79.

- Feltételezve, hogy egy rekord egy blokknyi méretű, mennyi az átlagos rekordelési idő?
- Tetszőleges másik hash függvényeket választva mennyi az elméletileg elérhető legjobb és legrosszabb véletlenszerű rekordelési idő ugyanekkora elemszámmal?

**3. feladat**

Vödörös hash-szervezéssel tárolunk egy állományt, amelyben a rekordok száma 15000. Egy rekord hossza 120 bájt, egy blokkba 4000 bájt fér el, egy kulcs hossza 25 bájt, egy mutatóé 8 bájt. A szervezést 10 vödörrel oldjuk meg. (Feltételezhetjük, hogy a hash-függvény egyenletesen osztja el a kulcsokat.)

- Mekkora az átlagos vödörméret?
- Mekkora diszkrterület szükséges a teljes struktúra tárolásához (valódi méret, illetve felhasznált tárterület)?
- Mennyi az átlagos rekordelési idő, ha a blokkelési idő 5 ms? (A keresés során a vödör-katalógust a memóriában tároljuk.)
- Mekkora legyen a vödrök minimális száma, ha a keresés során átlagosan 5 blokkelési idő alatt akarjuk megtalálni a keresett rekordot?

**4. feladat**

Egy állományt kétféle szervezéssel tudunk tárolni: sűrű index, majd erre épített egyszintes ritka index vagy pedig hash-algoritmussal. Az állományon néha intervallumkeresést is meg kell valósítani. Melyik szervezési módszert válasszuk? Adjon értelmes alsó becslést a szükséges blokkok számára az alábbi feltételek mellett:

- az állomány 3 000 000 rekordból áll
- egy rekord hossza 300 bájt
- egy blokk mérete 4000 bájt
- a kulchossz 45 bájt
- egy mutató hossza 5 bájt

**5. feladat**

Egy 10 000 000 rekordból álló állományt szeretnénk B\*-fa szervezéssel tárolni. A rekordhossz 850 bájt, egy blokk kapacitása (a fejrész nem számítva) 4000 bájt. A kulcs 50 bájtos, egy mutató tárolásához 18 bájt kell. Legalább hány blokkra van szükség? Mennyi az átlagos rekordelési idő, ha a memóriában egy blokk fér el? (Egy blokk elérésének ideje 5 ms.)

**Gondolkodtató kérdések**

- Elképzelhető-e sűrű index felhasználása hash-szervezés esetén?
- Mik a hash-szervezés előnyei, hátrányai a B\*-fával szemben?
- Milyen adatszerkezetet tudsz elképzelni egy térképszoftver adatbázisának?
- Milyen sorrendben kell beszúrunk a rekordokat egy B\*-fába ahhoz, hogy a legtöbb helyet pazaroljuk?
- Legfeljebb hány ritka index építhető közvetlenül egy heap-szervezésű állományra?
- Milyen plusz feladataink vannak beszúrásakor, illetve törléskor, ha sűrű indexek segítségével több B\*-fát építünk az adatbázisunkra?
- Lehet értelme egy kulcs szerint indexelt (B\*) adatbázis esetén is használni sűrű indexet? Mit nyerünk vele, és mennyit? Mitől függ, hogy mennyit nyerünk?
- Miért építünk B\*-fát, ha ritka indexszel is logaritmikus a keresés?
- Milyen nehézségeink adódnak, ha a töredékblokkokat is fel szeretnénk használni a merevlemezen?
- Miben különbözik egy kicsi és egy nagy blokkméretű lemezen tárolt adatbázis?
- Miért nem beszéltünk arról, hogy blokkon belül hogyan tároljuk az adatokat?
- Helyezd el a következő kifejezéseket a táblázatban: „blokknyi” „egyetlen”

	hány rekordot jelöl egy bejegyzése?	hány rekordot jelöl ki egy pointer-érték?
sűrű index		
ritka index		

- Hogyan változnának meg az adatbázisok, ha a jövőben a fizikai memóriában (az operatív tárban) helyezkedne el az adatbázisunk?

miértbani gyűjtés végéről a táblázatokról:

$$\{ITK, CMPD, TOSCM\} \quad \bar{T} = \{T \rightarrow \text{OSCM}, CM \rightarrow PD, IT \rightarrow K\}$$

	I	T	K	O	S	C	M	P	D	
ITK	a	a	a	$k_1^a$	$k_1^a$	$k_1^a$	$k_1^a$	$k_1^a$	$k_1^a$	$k_1^a$
CMPD	$b_2$	$b_2$	$b_2$	$b_2$	$b_2$	$b_2$	a	a	a	a
TOSCM	$b_3$	a	$b_3$	a	a	a	$k_3^a$	$k_3^a$		

} ez a végé

### Fizikai adatszervezés

diszkrétizálás klasszi  $\Rightarrow$  erre optimalizálunk

alapgyűjtés:  $b_1, b_2$

előbb  
utóbb

szűk index: pointer megadása  
nagy index: pointer klassza

szűk: ami alapján egy rekord egyértelműen azonosítható

$S_r$ : rekordméret

$n_r$ : rekordok száma

$k_r$ : blokkban lévő rekordok száma

$b_r$ : blokkok száma

$B$ : vödörök száma

$s$ : szűk mérete

$p$ : mutató mérete

$b$ : blokk mérete

1.)

$$n_r = 1000 \quad s_r = 850 \text{ byte} \quad b = 4000 \text{ byte} \quad k = 50 \quad p = 18$$

$$a) f_r = \left\lfloor \frac{b}{s_r} \right\rfloor = \left\lfloor \frac{4000}{850} \right\rfloor = 4 \text{ reboord / block}$$

$$b) b_r = \left\lfloor \frac{n_r}{f_r} \right\rfloor = 250 \text{ block}$$

$$f_i = \left\lfloor \frac{b}{k+p} \right\rfloor = \left\lfloor \frac{4000}{50+18} \right\rfloor = 58$$

$$b_i = \left\lfloor \frac{b_r}{f_i} \right\rfloor = 5 \text{ block} \quad \left. \vphantom{\frac{b_r}{f_i}} \right\} \text{indexallomány}$$

c) mita index: ~~minden~~ indexallomány rendelget

d) indexallomány 2 alkalom: 1 blockművelet

↳ 0H megtalálom, keresésem: 1 blockművelet

$$e) \left\lfloor \log_2 (5+1) \right\rfloor = 3 \text{ index} + 1 \text{ megtalált block} \text{ olvasása:}$$

$$4 \text{ olvasás} = 20 \text{ ms}$$

logy értékes  
eredményt  
kapzok

2.)

0	56, 91
1	36
2	79
3	52
4	
5	10
6	27

56, 36, 79, 52, 19, 27: 1 blockművelet

91: 2 blockművelet

$$\text{átlagosa} \quad 1 \cdot \frac{6}{7} + 2 \cdot \frac{1}{7} = 1,143$$

$$3.) n_r = 15000$$

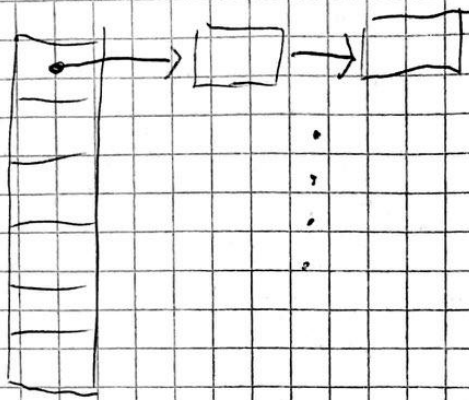
$$S_r = 120 \text{ byte}$$

$$L = 4000 \text{ byte}$$

$$L_2 = 25 \text{ byte}$$

$$p = 8 \text{ byte}$$

$$B = 10$$



$$a.) \left\lfloor \frac{n_r}{B} \right\rfloor = \left\lfloor \frac{15000}{10} \right\rfloor = 1500 \text{ rekord/vodor}$$

$$f_r = \left\lfloor \frac{L}{S_r} \right\rfloor = \left\lfloor \frac{4000}{120} \right\rfloor = 33 \text{ rekord/blokk}$$

$$\left\lfloor \frac{1500}{33} \right\rfloor = 46 \text{ db adatblokk egy vodorban}$$

b.)

$$\text{adat: } 15000 \cdot 120 \text{ byte} = 1800000 \text{ byte}$$

$$\text{használati tábla: } B \cdot p = 10 \cdot 8 \text{ byte} = 80 \text{ byte}$$

$$\text{összesen: } 1800080 \text{ byte}$$

felhasznált terület:

$$B \cdot 46 \text{ blokk} = 10 \cdot 46 = 460 \text{ blokk} = 460 \cdot 4000 = 1840000 \text{ byte}$$

$$\text{használati tábla: } 80 \text{ byte, de egy blokkot meg kell kördeni: } 4000 \text{ byte}$$

$$\text{összesen: } 1844000 \text{ byte}$$

c.) kezdő-tábla RM-ban

$$\left. \begin{array}{l} \text{kezdő eset: 1 előző művelet} \\ \text{legrosszabb eset: 46 előző művelet} \end{array} \right\} \text{átlag } \frac{1+46}{2} = 23,5 \text{ elem.}$$

117,5 ms

d.) 23,5 helyett 5:

$$\frac{1+x}{2} \leq 5 \quad \left. \vphantom{\frac{1+x}{2}} \right\} x=9$$

$$f_v = 33$$

$$\left[ \frac{n_r}{g \cdot f_v} \right] = \left[ \frac{15000}{9 \cdot 33} \right] = 5 \text{ db vödör kellene mozga}$$

lapon nem #, ezt kérés szerint beírjuk, hogy oldandó-e?  
feladat sűrű index, amit 2-vel indexel indexel

$$n_r = 1000$$

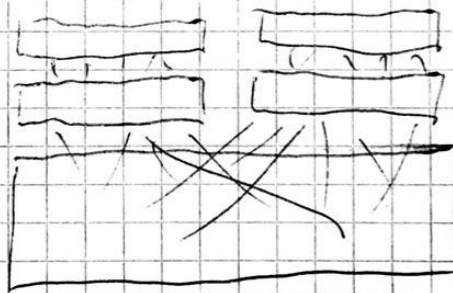
$$S_0 = 120 \text{ byte}$$

$$b = 1000 \text{ byte}$$

$$k = 90 \text{ byte}$$

$$P = 20 \text{ byte}$$

melkora lemozdított sűrűsítés



$$f_r = \left[ \frac{1000}{120} \right] = 8 \text{ db részlet/leírás}$$

$$b_r = \left[ \frac{1000}{8} \right] = 125 \text{ elem} \rightarrow 125000 \text{ byte}$$

sürü index:  $\underbrace{\text{başlangıç} + \text{uzunluk}}$

$$s + p = 70 \text{ byte}$$

$$f_i = \left\lfloor \frac{1000}{70} \right\rfloor = 14 \text{ dk rekord/blokk}$$

$$\left\lfloor \frac{1000}{14} \right\rfloor = 72 \text{ blokk} \rightarrow 72000 \text{ byte}$$

$$\left\lfloor \frac{72}{14} \right\rfloor = 6 \text{ blokk} \rightarrow 6000 \text{ byte}$$

$$\underbrace{125000}_{\text{adattörmény}} + \underbrace{2 \cdot 72000}_{\text{sürü indexek}} + \underbrace{2 \cdot 6000}_{\text{ritka indexek}} = 281000 \text{ bajt}$$

4., hasonlóan új rekord csomagok, a maradék a sürü indexeké.

Számold ki ottlan: a végén 268 739 blokk

5.)

$$n_r = 10000000$$

$$s_r = 850 \text{ byte}$$

$$b = 4000 \text{ byte}$$

$$s = 50 \text{ byte}$$

$$p = 18 \text{ byte}$$

$$f_r = \left\lfloor \frac{4000}{850} \right\rfloor = 4 \frac{\text{rekord}}{\text{blokk}}$$

$$b_r = \left\lfloor \frac{10000000}{4} \right\rfloor = 2500000 \text{ blokk}$$

$$f_i = \left\lfloor \frac{b + s}{s + p} \right\rfloor = 59$$

~~$$f_i = \left\lfloor \frac{b_r}{s + p} \right\rfloor = 58$$~~

$$\boxed{p \mid s \mid p \mid s \mid p \mid \dots \mid p \mid s \mid p \mid s \mid p}$$

Szűrt rekordok  
közvetlen tárolás

(9)

$$1. \text{ szint: } \left\lceil \frac{b_r}{4i} \right\rceil = \left\lceil \frac{2500000}{59} \right\rceil = 42373 \text{ blokk (adatállomány indexelése)}$$

$$2. \text{ szint: } \left\lceil \frac{42373}{59} \right\rceil = 719 \text{ blokk}; \quad 3. \text{ szint: } \left\lceil \frac{719}{59} \right\rceil = 13 \text{ blokk}$$

$$4. \text{ szint: } \left\lceil \frac{13}{59} \right\rceil = 1 \text{ blokk}$$



4 szintes B\* fa,

összesen:

$$42373 + 719 + 13 + 1 =$$

$$= 43106 \text{ blokk}$$

innen az adatok  $4 + 1$  blokkok mélységig elérhetők. ritka indexelés volna; 13 színelmére.

$$4. \text{ szintig érő számanak számolása: } \left\lceil \log_{59} 2500000 \right\rceil = 4$$



## 1. feladat: ismerkedés a jelölésekkel, katalógusinformációk, algoritmusok használata

Egy bank nyilvántartásában található, SZÁMLA(TELEPÜLÉS, EGYENLEG, ...) sémára illeszkedő relációból szeretnénk megtudni a budapesti számlák adatait. Ezeket tudjuk a relációról:

- $f_{\text{SZÁMLA}} = 40$ : SZÁMLA reláció 40 rekordja fér bele egy lemezblokkba.
- $V(\text{TELEPÜLÉS, SZÁMLA}) = 50$ : 50 különböző fiók-név létezik a SZÁMLA relációban.
- $V(\text{EGYENLEG, SZÁMLA}) = 500$ : 500 különböző értékű számla van a SZÁMLA relációban.
- $n_{\text{SZÁMLA}} = 10\ 000$ : A SZÁMLA relációnak 10 000 eleme van.

A feladat megoldása során tegyük fel, hogy a reláció elemeit optimális tárkihasználás mellett tároljuk. Kérdések:

- (a) Adjuk meg a feladatot megoldó relációs algebrai lekérdezést!
- (b) Mennyi minimális/maximális/átlagos költség, ha lineáris keresést alkalmazunk? Mitől függ, hogy mennyi?
- (c) Tfh. a rekordok a fiók szerint rendezetten tárolódnak. Mennyi a bináris keresés várható költsége?

## 2. feladat: a join nagyságának becslése

Adott két relációs sémánk, a BANKBETÉT és az ÜGYFÉL. Illesszük a két (sémára illeszkedő) relációt a mindkettőben szereplő ÜGYFÉL\_NÉV attribútum szerint, amely az ÜGYFÉL kulcsa, a BANKBETÉT-ben pedig idegen kulcs. Tegyük fel, hogy a két relációról a következő katalógusinformációk állnak rendelkezésre:

- $n_{\text{ÜGYFÉL}} = 10\ 000$
- $f_{\text{ÜGYFÉL}} = 25$  ( $b_{\text{ÜGYFÉL}} = ?$ )
- $n_{\text{BANKBETÉT}} = 5\ 000$
- $f_{\text{BANKBETÉT}} = 50$ . ( $b_{\text{BANKBETÉT}} = ?$ )
- $V(\text{ÜGYFÉL\_NÉV, BANKBETÉT}) = 2\ 500$ , ( $SC(\text{ÜGYFÉL\_NÉV, BANKBETÉT}) = ?$ )

Mekkora a BANKBETÉT és az ÜGYFÉL természetes illesztésének mérete, ha egyetlen közös attribútumuk az ÜGYFÉL\_NÉV?

Általánosítsuk a feladatot az alábbi esetekre ( $R$  és  $S$  az illesztendő relációk sémái, természetes illesztéssel!)

- $R \cap S = \emptyset$ .
- $R \cap S$  az  $R$  reláció kulcsa.
- $R \cap S \neq \emptyset$  egyik relációs sémának sem kulcsa.

## 3. feladat: hash join költsége

Számítsd ki a „hashjoin” algoritmussal végrehajtott join költségét, ha vödörös hashelést alkalmazunk! A hash függvény egyenletes eloszlással képezi le a kulcsokat az értékészletére. Hogyan érdemes a join-t végrehajtani? A blokkméret nettó 2 000 bájt, a hashtábla szokás szerint elfér a memóriában.

- $R$  reláció: 120 000 rekord, rekordhossz 150 bájt, kulcs 12 bájt, mutató 8 bájt, a hash tábla mérete 10 000 bájt.
- $S$  reláció: 10 000 rekord, rekordhossz 250 bájt, kulcs 15 bájt, mutató 8 bájt a hash tábla mérete 1 000 bájt.

## 4. feladat: index-alapú illesztés

Számítsd ki az illesztés költségét, ha elsődleges, B\*-fa struktúrájú indexeket használhatunk a join attribútumok szerinti rekordelérésre! A blokkméret nettó 4 000 bájt. Melyik reláció legyen a külső hurokban? Hányszoros válaszdöt kapunk, ha az optimalizáló rosszul dönt?

- $R$  reláció: 140 000 rekord, rekordhossz 140 bájt, kulcs 10 bájt, mutató 4 bájt.
- $S$  reláció: 15 000 rekord, rekordhossz 300 bájt, kulcs 6 bájt, mutató 4 bájt.

## Gondolkodtató kérdések

1. Relációanalízis feladat: A relációs lekérdezések költség alapú optimalizálásakor az ekvivalens alakokat előállító szabályoknak nincs jelentősége, mert csak a heurisztikus (vagy szabály alapú) optimalizálás alapszik átalakítási szabályokon.
2. Vizsgáld meg a blokkalapú egymásba ágyazott ciklikus illesztés és az egymásba ágyazott ciklikus illesztés algoritmusokban: a legbelső ciklusmag hányszor fut le worst-case esetben. Vesd össze az eredményt a két algoritmus worst-case költségével, és magyarázd meg a tapasztaltakat.
3. A természetes illesztésnél előbb „érdemes” végrehajtani az egy relációt érintő szelekció műveleteket, és ezt meg is tehetjük: ekvivalens algebrai alakot kapunk, ha lentebb (az alaprelációk irányába) süllyesztjük a szelekciókat. Megtehető-e ugyanez külső illesztés esetén? Mitől függ, hogy megtehető-e?
4. Lehet-e értelme olyan alapos költségbecslést és optimalizációt végezni, amelynek a költsége eléri/meghaladja a lekérdezés legköltségesebb szóba jövő végrehajtási módja költségét?

5. Melyek azok a műveletek, ahol elsődleges index használata olcsóbb végrehajtást eredményez, mint egy egyszerű index használata?
6. Mutass példát arra, amikor az elsődleges index használata ront a teljesítményen!
7. Mutass példát, amikor a lineáris keresés olcsóbb, mint az indexelt keresés? Próbáld általánosan megfogalmazni az eredményt!
8. Mi a biztos jele annak, hogy adatbázisunkból a szekvenciálisan generált ID nevű, elsődleges kulcsmezőt törölni kell? Mit veszünk azzal, ha nem töröljük?
9. A relációs lekérdezések végrehajtási folyamatában szereplő optimalizálás bemenete miért relációs algebrajellegű kifejezés, ahelyett hogy valamilyen kalkulus-kifejezés lenne?
10. Az „index alapú illesztés” feladatban a relációs sémák kulcsait különböző hosszon tároltuk, azonban a természetes illesztés miatt rajtuk egyenlőségvizsgálatot kellett végezni. Lehetséges ilyen eset? Mutass példát rá, ha lehetséges, és indokold, ha nem!
11. Az „index alapú illesztés” feladatban erre vonatkozó információ hiányában feltételeztük, hogy az indexelt kulcsok egyedi értékűek a sémára illeszkedő relációban. Hogyan változik a join költsége/memóriaigénye, ha az egyes relációkban
  - $SC(Kulcs, R) < f_R$ ?
  - $SC(Kulcs, R)$  tetszőleges?

### Gondolkodtató példa: több attribútumos indexek

A „D” Bank napi működését támogató adatbázisban a tranzakciókat a TRANZAKCIÓ(ÜGYFÉLSZÁM, DÁTUM, IDŐPONT, TÍPUS, PARTNER\_SZÁMLASZÁMA, ÖSSZEG) sémára illeszkedő, heap szervezésű relációban tárolják. A tranzakció-rekordok nem módosíthatók. A napi működés középpontjában az ügyfél áll: a banki tranzakciók, mint pl. a vásárlás, átutalás, ügyfélmúlt lekérdezése (pl. netbankon max. 90 napra vonatkozóan) hozzá kapcsolódnak, ezért a működést az (ÜGYFÉLSZÁM, DÁTUM) attribútum párra épített elsődleges, B\*-fa struktúrájú index támogatja.

Elemzéshez a bank indulásától fogva előállítják az adatbázis egy másolatát, melyben a tárolási struktúra megegyezik az on-line rendszerrel, de az új rekordok betöltése napi szinten történik: mindig az előző napi új rekordok kerülnek áttöltésre. Az első időszakban flottul ment a dolog: a hajnali 3:03 órakor elindított áttöltés pillanatok alatt lefutott. Az első üzleti év végén azt tapasztalták, hogy 7 óra körül készül el az áttöltés. Fél év múlva csak délelőtt 11 óra után állt készen az elemzőrendszer. Az elemzési osztály vezetője egy tankönyvet lobogtatva ment a vezérigazgatóhoz, amiben azt olvasta (környezetéből kiragadott idézet): „a nagy mennyiségű adat batch jellegű betöltését tipikusan jelentősen lassítja, ha index van a cél táblán”. Nosza, a vezérigazgató kiutalta a prémiumot az osztályvezetőnek, és utasította az informatikusokat az elemzőrendszerbeli index lekapcsolására.

A bank ügyfélköre kezdettől fogva egyenletesen emelkedett az első év végéig, amikor megállt a növekedés: pontosan 160 000 ügyfele van a banknak. Az ügyfél-lemorzsolódás elhanyagolható. Egy tipikus ügyfél naponta 4 tranzakciót végez. A blokkméret nettó 4 000 bájt, egy tranzakciós rekord 100 bájt, amiből az ügyfélszám 8 bájt, a dátum 4 bájt. A rendszer 64 bites mutatókat használ. Egy blokkművelet időigénye 5 ms.

- Milyen hatással van az index lekapcsolása az elemzők által futtatott lekérdezésekre, amelyek tipikusan 1 havi összesítő számításokat végeznek? (azaz mennyi idő alatt futott le az index lekapcsolása előtt egy ilyen lekérdezés, és hogyan változott az időigény az index lekapcsolása után)
- Számítsuk ki, vajon segít-e az elemzőrendszerbeli index lekapcsolása a bank áttöltési problémáján.
- Javasoljunk jobb megoldást a bank problémájára, és számítsuk ki, hogy változatlan ügyfélszám és adatbázis-kezelő rendszer mellett hány évig biztosítja a megoldás, hogy 7 óra körül készen álljon az áttöltés?
- Most, hogy a korszakalkotó javaslat után az elemzőrendszer mindig készen áll a reggeli munkakezdésre, optimalizáljuk azt is a korábban említett lekérdezési profilra úgy, hogy ne romoljon jelentősen az áttöltési teljesítmény!

november 14.

# Lebonyozás optimalizáció

valamilyen deklaratív  
nyelvezésből (pl. SQL)

→ épít egy kátl,  
ami valamilyen  
relációs algebrahoz  
masonlít

végrehajtási  
terv

← let optimalizálógia

↳ tipikusan 10  
szempontjából

jelöléses:

$n_r$  = rekordok száma

$b_r$  = felhasznált blokkok száma

$S_r$  = rekord méret

$f_r$  = rekord/blokk (blocking factor)

$V(A, v) = |\pi_A(v)|$  (kardinalitás, hány különböző értékbe  
került A-oszlop)

$SC(A, v) = \frac{n_r}{V(A, v)}$

$f_i$  = index rekord/blokk

$HT_i$  = B\* fájl a szintek száma

$LB_i$  = legalsó szint ~~(#)~~ blokkok száma

vannak itt mindenféle algoritmusok, amik egyenlőre nem teljesen,  
de vizsgálva tudni kell persze.

1.)

$$f_{\text{számla}} = 40 \text{ rekord/blokk}$$

$$v(\text{település}, \text{számla}) = 50$$

$$v(\text{egység}, \text{számla}) = 500$$

$$n_{\text{számla}} = 10.000$$

a.)  $\sigma_{\text{település} = \text{'Budapest'}}(\text{számla})$

$$b.) k_{\text{számla}} = \left\lceil \frac{n_{\text{számla}}}{f_{\text{számla}}} \right\rceil = \left\lceil \frac{10.000}{40} \right\rceil = 250 \text{ blokk}$$

miért nem rendezett? mindig is 250

mi van, ha rendezett? (válasz szerint)

$$\frac{k_{\text{számla}} + 1}{2} + \left\lceil \frac{S(\text{település}, \text{számla})}{f_{\text{számla}}} \right\rceil - 1$$

első budapesti  
entés megtalálása

az első né  
legyen 2x  
számla

$$S(\text{település}, \text{számla}) = \frac{n_{\text{számla}}}{v(\text{település}, \text{számla})} = \frac{10.000}{50} = 200$$

$$125,5 + 5 - 1 = 129,5$$

$$c.) \left\lceil \log_2 (k_r + 1) \right\rceil + \left\lceil \frac{SC(A, v)}{f_r} \right\rceil - 1$$

$$\left\lceil \log_2 (260 + 1) \right\rceil + \frac{200}{40} - 1 = 8 + 5 - 1 = 12$$

2.)

$$n_{\text{ügyfél}} = 10.000$$

$$f_{\text{ügyfél}} = 25 \rightarrow k_{\text{ügyfél}} = \frac{10000}{25} = 400$$

$$n_{\text{bankbetét}} = 5.000$$

$$f_{\text{bankbetét}} = 50 \rightarrow k_{\text{bankbetét}} = \frac{5.000}{50} = 100$$

$$V(\text{ügyfél-név, bankbetét}) = 2500$$

egyetlen közös attribútum: ügyfél-név

$$n_{\text{illesztés}} = n_{\text{bankbetét}} = 5000$$

azért, mert ennél több nem lehet, ill. már ennyi listázás lesz, mint a maximálisan lehetséges

$$R \cap S = \emptyset \text{ ebben az esetben } n_{\text{illesztés}} = n_r \cdot n_s$$

$$R \cap S = R \text{ kölcsön} \rightarrow n_{\text{illesztés}} = n_s$$

$$R \cap S = \emptyset \text{ egyik személynél nem kölcsön} \rightarrow R \cap S = \emptyset$$

$$SC(w, s) = \frac{n_s}{V(w, s)}$$

$$n_v \cdot SC(w, s) = \frac{n_s \cdot n_r}{V(w, s)}$$

$$n_s \cdot S(w, r) = \frac{n_s \cdot n_r}{V(w, r)}$$

(2)

3y

$$k_r + n_r \cdot c$$

1. reláció

$$\text{kash-talbla beszerzésének száma: } \frac{10.000}{8} = 1250 \quad \begin{array}{l} \text{ennyi vödör} \\ \text{van} \end{array}$$

$$f_1 = \left\lfloor \frac{2000}{150} \right\rfloor = 13 \text{ reborod}$$

$$\frac{120.000}{1250} = 96 \rightarrow \left\lfloor \frac{96}{13} \right\rfloor = 8 \text{ blokk}$$

elgrosztabb esetben  
ennyit kell olvasni

$$c_1 = \frac{1+8}{2} = 4,5 \rightarrow \text{átlag}$$

2. reláció

$$\text{kash-beszerzésének száma: } \frac{1000}{8} = 125 \text{ db vödör}$$

$$f_2 = \left\lfloor \frac{2000}{250} \right\rfloor = 8 \text{ reborod}$$

$$\frac{10.000}{125} = 80 \quad \begin{array}{l} \text{reborod} \\ \text{vödör} \end{array}$$

$$\left\lfloor \frac{80}{8} \right\rfloor = 10 \quad \begin{array}{l} \text{blokk} \\ \text{vödör} \end{array} \rightarrow c_2 = \frac{1+10}{2} = 5,5 \text{ átlag}$$

$$\text{Költség 1} = 8 \cdot 1250 + 120.000 \cdot 5,5$$

$$\text{Költség 2} = 10 \cdot 125 + 10.000 \cdot 4,5$$

4.)

$$b = 4000 \text{ byte}$$

1. reláció

$$f_1 = \left\lfloor \frac{4000}{140} \right\rfloor = 28 \text{ rekord/blokk} ; b_1 = \left\lceil \frac{140 \cdot 000}{286} \right\rceil = 5000 \text{ blokk}$$

$$f_{i1} = \left\lfloor \frac{4000}{14} \right\rfloor = 285 \text{ indexrekord/blokk}$$

$$\left\lfloor \frac{b+s}{s+p} \right\rfloor$$

→ 286 indexrekord/blokk

2 szintű a fa (HT<sub>i1</sub> = 2)

$$LB_{i1} = \left\lceil \frac{5000}{286} \right\rceil = 18$$

2. reláció

$$f_2 = \left\lfloor \frac{4000}{800} \right\rfloor = 5 \text{ rekord/blokk} ; b_2 = \left\lceil \frac{15 \cdot 000}{13} \right\rceil = 1154 \text{ blokk}$$

$$f_{i2} = \left\lfloor \frac{4000}{6+4} \right\rfloor = 400$$

$$\left\lfloor \frac{b+s}{s+p} \right\rfloor$$

→ 400 indexrekord/blokk

2 szintű a fa (HT<sub>i2</sub> = 2)

$$LB_{i2} = \left\lceil \frac{1154}{400} \right\rceil = 3 \text{ blokk}$$

$$S_{\text{öss}}(R_1) = 5000 + 140 \cdot 000 \cdot (2+1)$$

$$S_{\text{öss}}(R_2) = 1154 + 15 \cdot 000 \cdot (2+1)$$

reklamálták ki, hogy elsőleges index, így rendezett az egész:

$$b_1 + b_2 = 5000 + 1154 = 6154$$



**1. feladat:** Legális-e az 1a. táblázat által mutatott ütemezés? Ha nem, mit kellene javítani rajta, hogy azzá váljon?

**2. feladat:** Ellenőrizd, hogy a 1c. táblázaton látható ütemezés legális-e! Rajzold meg a sorosíthatósági gráfot, dönts el, hogy sorosítható-e az ütemezés! Ha igen, adj egy soros ekvivalenst, ha nem, mutasd meg, miért nem! Hogy nézne ki a gráf, ha egyszerű zármodellt használnánk?

**3. feladat:** Legális-e az 1b. táblázat szerinti ütemezés? A tranzakciók követik-e a 2PL-t? Hol van a tranzakciók zárpontja? Mi egy soros ekvivalens ütemezés?

**4. feladat:** Időbélyeges tranzakciókezelést használunk R/W modellben. Jegyezd fel az alábbi sorozat minden művelete után az R(A), R(B), W(A), W(B) értékeit, ha kezdetben mindegyik 0. Mely tranzakciók abortálnak?  $r_i$  és  $w_i$  a  $T_i$  tranzakció olvasás (r) és írás (w) műveleteit jelöli, és  $t(T_i) = i$ .

$$r_1(A), r_2(B), r_1(B), w_3(B), r_2(B), w_4(A), r_4(B), w_1(A), w_3(B)$$

**5. feladat:** Oldd meg az 5. feladatot, verziókezeléssel kiegészítve! Most mi történik?

**6. feladat:** Egy rendszerleállítás után a napló vége a 1d. táblázat szerinti bejegyzéseket tartalmazza. Melyek a redo helyreállítás lépései? Mi lesz a helyreállítás után A, B és C értéke?

### Gondolkodtató kérdések

**1:** A naplózás tárhelyigényét szeretnénk optimalizálni. Helyes-e a következő érvelés? Egy tranzakciónak csak a COMMIT pontjáig van szüksége a naplóra – hiszen a COMMIT utáni műveletek biztosan lefutnak –, ezért szigorú 2PL alkalmazásával megelőzzük a lavinahatást, és a COMMIT naplózása helyett így a naplóból már törölhetjük az adott tranzakcióhoz tartozó bejegyzéseket (ha garantáljuk ezen törlés atomicitását).

**2:** Hogyan biztosítja a holtpontmentességet a 2PL?

**3:** Igaz-e, hogy egy kétfázisú protokoll estén a tranzakciók mindig helyesen futnak le? (Mit jelenthet az, hogy "helyesen"?)

**4:** Lehet-e konkurrensen módosítani egy állományt, amire B\* fa épül? Mikor lehet felszabadítani a gyökérelemet fogó zárat?

**5:** Miért fontos a sorosíthatóság?

**6:** Egy ütemezés nem sorosítható. Ennek ellenére érvényes lehet-e az izolációs elv?

**7:** Ha a naplófájl tartalmaz minden információt a változásokról, akkor miért kell az adatbázis?

**8:** Mondj példát kézenfekvő soros ekvivalensre 2PL és időbélyeges tranzakciókezelés esetén!

**9:** Mikor érdemes 2PL-t és mikor időbélyeges tranzakciókezelést alkalmazni?

**10:** Mi történik a sorosíthatósági gráffal, ha egy tranzakció abortál?

**11:** Az időbélyeges tranzakciókezelés miként véd a holtpont ellen?

**12:** Mondj példát vagy ellenpéldát a következő esetekre!

(a) Időbélyeges tranzakciókezelés esetén egy tranzakció READ során abortál.

(b) Időbélyeges tranzakciókezelés esetén egy tranzakció WRITE során abortál.

(c) Időbélyeges tranzakciókezelést verziókezeléssel együtt alkalmazunk. Egy tranzakció READ során abortál.

(d) Időbélyeges tranzakciókezelést verziókezeléssel együtt alkalmazunk. Egy tranzakció WRITE során abortál.

**13:** Hogyan tároljuk az időbélyegeket? Mit tudunk mondani a számukra vonatkozóan? Meddig kell fenntartani?

### Kiegészítő feladatok

**1. feladat:** Ellenőrizd, hogy a 2. táblázaton látható ütemezés legális-e! Rajzold meg a sorosíthatósági gráfot, dönts el, hogy sorosítható-e az ütemezés! Ha igen, adj egy soros ekvivalenst, ha nem, mutasd meg, miért nem! Hogy nézne ki a gráf, ha RLOCK-WLOCK modellt használnánk? (Ahol csak olvasunk, ott RLOCK-kal, ahol írunk (is) ott WLOCK-kal helyettesítjük értelemszerűen a LOCK-ot.)

**2. feladat:** Egy tranzakció az 1. ábra által mutatott hierarchiában a D, I és J adategységekhez szeretne hozzáférni, majd befejezné működését. Milyen zárat kell és hova elhelyeznie, illetve milyen sorrendben kell felszabadítania a zármenedzsernek, ha mindenkor a lehető legkevesebb zárat szeretné fenntartani.

	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>
(1)			RLOCK B
(2)			READ B
(3)		WLOCK B	
(4)			RLOCK A
(5)	RLOCK A		
(6)		WRITE B	
(7)			READ A
(8)	READ A		
(9)	UNLOCK A		
(10)			UNLOCK B
(11)			UNLOCK A

1a. táblázat

	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>
(1)	LOCK A		
(2)		LOCK B	
(3)			LOCK C
(4)			LOCK D
(5)	LOCK E		
(6)	UNLOCK A		
(7)			UNLOCK D
(8)		LOCK A	
(9)		LOCK D	
(10)	UNLOCK E		
(11)		UNLOCK B	
(12)			UNLOCK C
(13)		UNLOCK A	
(14)		UNLOCK D	

1b. táblázat

checkpoint
(T <sub>1</sub> , begin)
(T <sub>2</sub> , begin)
(T <sub>2</sub> , A, 20)
(T <sub>2</sub> , B, 10)
(T <sub>1</sub> , A, 2)
(T <sub>3</sub> , begin)
(T <sub>1</sub> , C, 5)
(T <sub>1</sub> , commit)
(T <sub>3</sub> , C, 6)
(T <sub>3</sub> , commit)

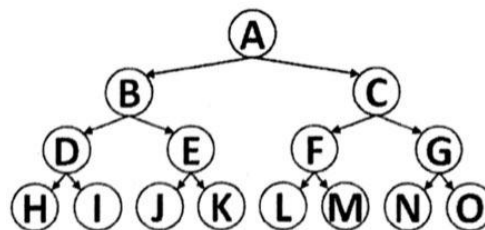
1d. táblázat

	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>
(1)		RLOCK F		
(2)	RLOCK A			
(3)			RLOCK D	
(4)			UNLOCK D	
(5)	UNLOCK A			
(6)			WLOCK B	
(7)	RLOCK D			
(8)				RLOCK A
(9)	WLOCK E			
(10)				UNLOCK A
(11)			UNLOCK B	
(12)				WLOCK A
(13)	UNLOCK D			
(14)	RLOCK C			
(15)				UNLOCK A
(16)		UNLOCK F		
(17)			WLOCK A	
(18)		WLOCK B		
(19)	UNLOCK E			
(20)	UNLOCK C			
(21)				RLOCK D
(22)				UNLOCK D
(23)			WLOCK C	
(24)			UNLOCK C	
(25)		WLOCK D		
(26)			UNLOCK A	
(27)		RLOCK E		
(28)		UNLOCK E		
(29)				RLOCK E
(30)		UNLOCK B		
(31)	RLOCK F			
(32)		UNLOCK D		
(33)	UNLOCK F			
(34)				UNLOCK E

1c. táblázat

	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>
(1)	LOCK A			
(2)		LOCK B		
(3)	READ A			
(4)	UNLOCK A			
(5)		WRITE B		
(6)		LOCK C		
(7)			LOCK A	
(8)		READ C		
(9)			READ A	
(10)		UNLOCK C		
(11)			UNLOCK A	
(12)			LOCK C	
(13)		UNLOCK B		
(14)			READ C	
(15)	LOCK A			
(16)				LOCK B
(17)	READ A			
(18)				READ B
(19)			UNLOCK C	
(20)				UNLOCK B
(21)			LOCK B	
(22)	WRITE A			
(23)	UNLOCK A			
(24)			READ B	
(25)			UNLOCK B	
(26)		LOCK A		
(27)				LOCK B
(28)		WRITE A		
(29)				WRITE B
(30)		UNLOCK A		
(31)				UNLOCK B

2. táblázat



1. ábra

november 28.

● elmeletti gyorstatasok:

egy lekkozasnal: szoros

az toz so, sorozatossag  
összevissza mindent

szorositate  
ütemezés  
(sorozat)

megoldasok:

- zárolás <sup>egyszerű</sup> ← R/W

- időbeliség

- szabvány protokoll, ami

biztosítja a szorositatekat  
(2PL)

● ha valaki abortalend,  
akkor a rd varozat is  
szell (lavina)

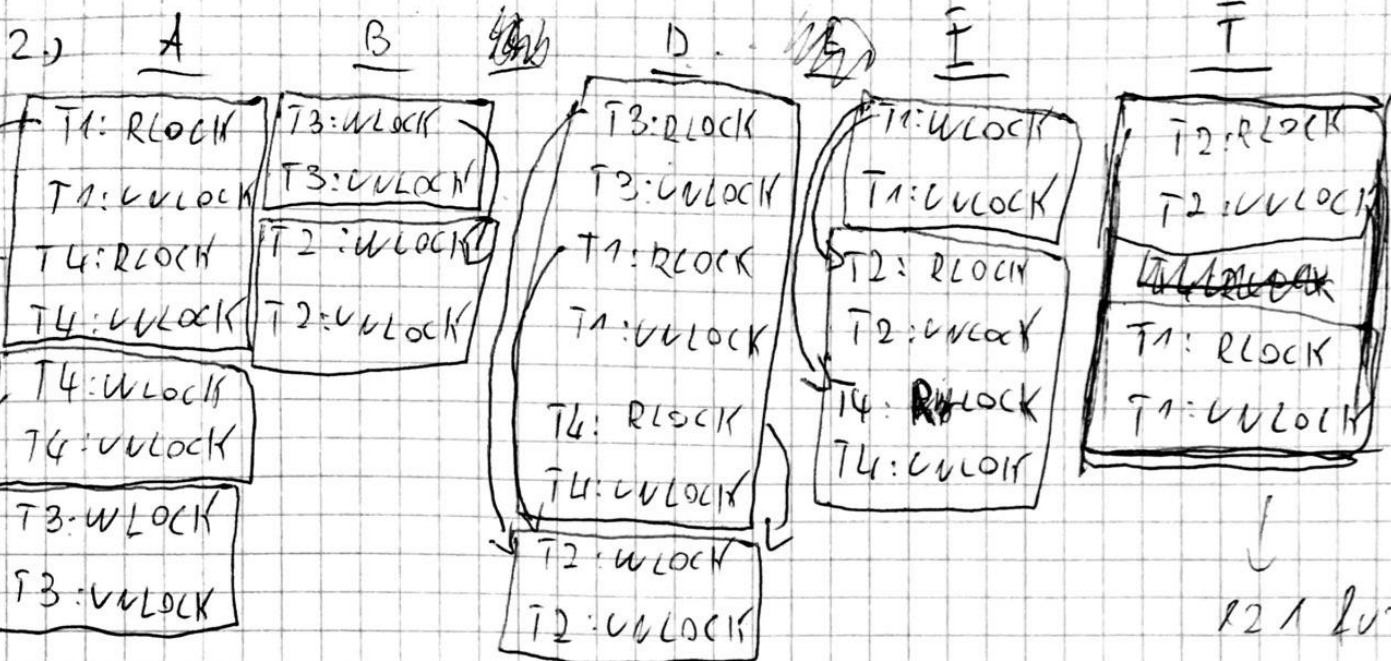
1.) A: T3: RLOCK  
 T1: RLOCK  
 T3: READ  
 T1: READ  
 T1: UNLOCK  
 T3: UNLOCK

B: T3: RLOCK  
 T3: READ  
 T2: WLOCK  
 T2: WRITE  
 T3: UNLOCK

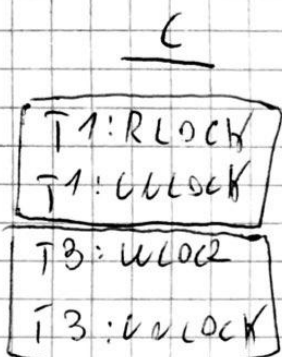
↳ zárolás  
↓ kompatibilis

↳ T2 nem  
UNLOCK-ol

megoldas: T3 unlock B-jet folytatja lefectne teni,  
T2-t meg nem elfolasztani, unlockolni.



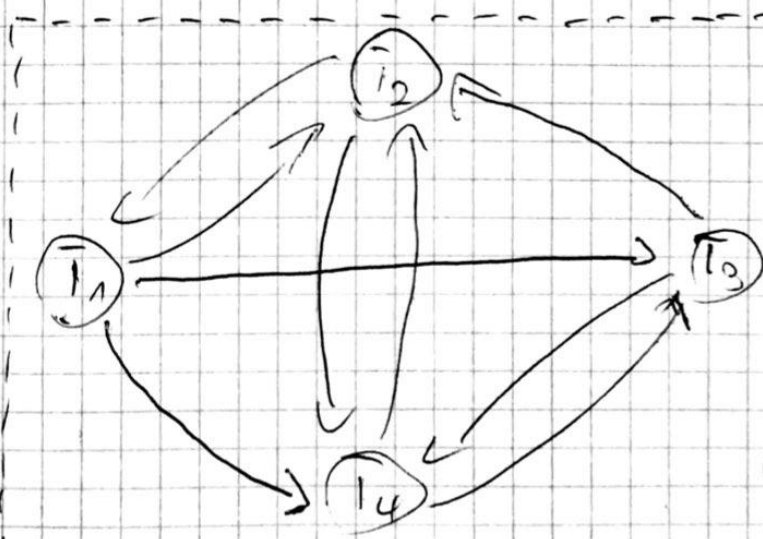
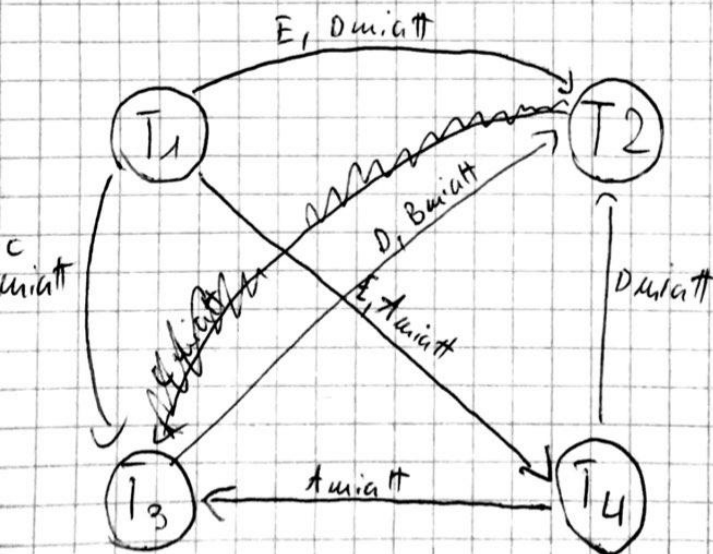
↓  
121 futam



- RW: olvasásból a w-ra
- RR: olvasásból a másolásra
- WB: olvasásból a r-re

és akkor elbír szigor, hogy ez legalább.

lock futam - algoritmus graf. megfogalmazása



3. ~ ugyanazt kell csinálni, mint az előbb.  
 egyszerűen az adatokban működik a rendezéses módszer is.

<u>A</u>	<u>B</u>	<u>D</u>	<u>E</u>
T1: LOCK	T2: LOCK	T3: LOCK	T1: LOCK
T1: UNLOCK	T2: UNLOCK	T3: UNLOCK	T1: UNLOCK
T2: LOCK		T2: LOCK	
T2: UNLOCK		T2: UNLOCK	

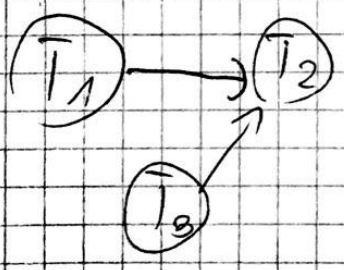
C

T3: LOCK  
 T3: UNLOCK

✓  
 logalis

Zárpontok: T1: 5.000  
 T2: 0.500  
 T3: 4.000

és az elvárások: pl.: T3 → T1 → T2 v.  
 T1 → T3 → T2



Éggyél felre a záradék, de legrosszabb időkezeléssel.

	T olvasni	T írni
$t(T) < R(A)$	abort	abort
$t(T) < W(A)$	abort	abort
$t(T) < R(A)$	✓ $R(A)$ nem változik	abort
$t(T) > W(A)$	abort	abort
$t(T) > R(A)$	abort	abort
$t(T) < W(A)$	✓ $R(A) := t(T)$	✓ $W(A) := t(T)$
$t(T) > R(A)$		
$t(T) > W(A)$		

szabályok,  
nagyjából logikusak,  
sokkal egyszerűbbek

	R(A)	W(A)	R(B)	W(B)
-	0	0	0	0
$r_1(A)$	1	0	0	0
$r_2(B)$	1	0	2	0
$w_1(B)$	1	0	2	3
$w_2(B)$	1	0	2	3
$w_3(B)$	1	0	abort T2	3
$w_4(A)$	1	4	2	3
$r_4(B)$	1	4	4	3
$w_1(A)$	1	abort T1	4	3
$w_2(B)$	1	4	4	3 abort T3 abort T4

6. naplózás: durability-hoz, módosításokat logoljuk  
 checkpoint: legutóbbi konzisztens állapot

azokat kell rögzíteni, amikbe commitolva vették.

$f = 2$

$C = 6$

$B = ?$

5.) 4-es feladat + verziókezelés

	A0	f1	A2	B0	B1
$r_1(A)$	1/0	-	-	0/0	-
$r_2(B)$	1/0	-	-	2/0	-
$r_1(B)$	1/0	-	-	2/0	-
$w_2(B)$	1/0	-	-	2/0	2/3
$r_2(B)$	1/0	-	-	2/0	2/3
$w_4(A)$	1/0	1/4	-	2/0	2/3
$r_4(B)$	1/0	1/4	-	2/0	4/3
$w_1(A)$	1/0	1/4	1/1	2/0	4/3
$w_3(B)$					

abort: 4-es mártólcsatt -> 3-as jelen írást  
 T3 abort -> T4 abort (8. sor)