Adatbázisok elméleti alapjai

Dr. Kiss Attila

people.inf.elte.hu/kiss

kiss@ullman.inf.elte.hu

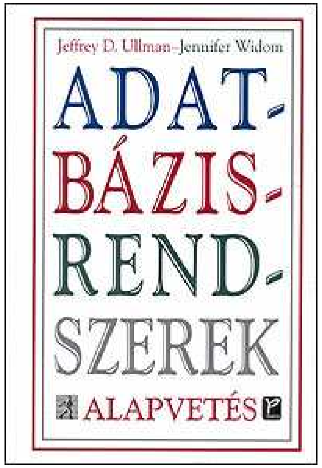
D.2.508

Tematika

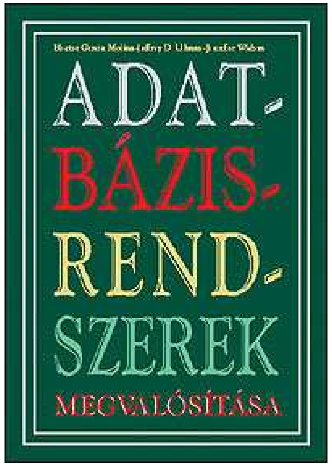
1. **Adatbázis-kezelő rendszerek** általános jellemzői.
2. **A relációs adatmodell**, a relációs algebra műveletei, használata, algebrai optimalizáció.
3. **Relációs kalkulusok**, DRC, TRC, tartományfüggetlenség, biztonságosság, lekérdező nyelvek ekvivalenciája, átírás egyik nyelvből a másikba.
4. **Az SQL nyelv részei** (ORACLE specifikusan):

   -DDL, DML QL, triggerek, jogosultságok, PL/SQL, függvények, procedúrák, cursorok használata, programozás, rekurzív SQL, datalog.
5. **Adatmodellezés**, egyed-kapcsolat modell, az E/K diagram átalakítása relációs adatmodellé.
6. **Relációs adatbázisok tervezése,** anomáliák, funkcionális és többértékű függőségek, implikációs probléma, attribútumhalmazok lezárása, dekompozíciók tulajdonságai, veszteségmentesség, függőségőrzés ellenőrzése, Boyce-Codd normálforma, 3NF, 4NF, dekomponáló algoritmusok.
7. **Fizikai fájlszervezés alapjai**, blokkok, kupac és hash szervezés, rendezés, elsődleges és másodlagos indexek, többszintű indexek, B^+-fák, B*-fák, katalógusok, indexelések, lekérdezési tervek az Oracle rendszerben.

IRODALOM



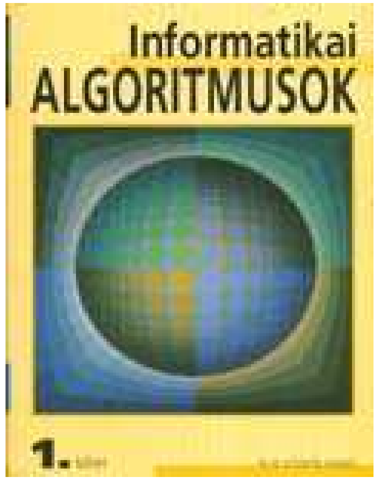
ABR1



ABR2

SQL

ORACLE



ALG1

Edgar Frank Codd 12 szabálya

1. **Az egységes megjelenésű információ szabálya**

   Az adatbázisban szereplő összes információt egy, és csak egy megadott formában (adatmodellben) lehet ábrázolni, nevezetesen táblázatok sorainak oszlopértékeiben.

2. **Garantált lokalizálhatóság szabálya**

   Az adatbázisban minden egyes skaláris értékre logikailag úgy kell hivatkozni, hogy megadjuk az azt tartalmazó táblázat és az oszlop nevét, valamint a megfelelő sor elsődleges kulcsának az értékét.

3. **A NULL értékek egységes kezelése**

   Az adatbázis-kezelő rendszernek (DBMS) olyan egységes módszerrel kell támogatnia a hiányzó vagy nem ismert információ kezelését, amely eltér az összes „rendes” érték kezelésétől, továbbá független az adattípustól.

4. **A relációs modell alapján aktív online katalógust kell üzemben tartani**

   A rendszernek támogatnia kell egy online, beépített katalógust, amelyet a feljogosított felhasználók a lekérdező nyelv segítségével ugyanúgy le tudnak kérdezni, mint a közönséges táblákat.

5. **A teljes körű „adatnyelv” szabálya**

   A rendszernek legalább egy olyan relációs nyelvet kell támogatnia, amelynek

   (a) lineáris a szintaxisa,

   (b) interaktívan és az alkalmazásokhoz készített programokon belül is lehet használni,

   (c) támogatja az adatdefiniáló műveleteket, a visszakereső és adatmódosító (manipulációs) műveleteket, biztonsági és jósági (integritási) korlátokat, valamint a tranzakciókezelési műveleteket (begin, commit, rollback: elkezdés, jóváhagyás és visszagörgetés).

6. **A nézetek frissítésének szabálya**

   A rendszernek képesnek kell lennie az adatok összes nézetének frissítésére.

Edgar Frank Codd 12 szabálya

7. Magas szintű beszúrás, frissítés és törlés

A rendszernek támogatnia kell az INSERT, UPDATE, és DELETE (új adat, módosítás, törlés) operátorok halmaz szintű, egyidejű működését.

8. Fizikai szintű adatfüggetlenség

A fizikai adatfüggetlenség akkor áll fenn, ha az alkalmazások (programok) és a felhasználók adatelérési módja független az adatok tényleges (fizikai) tárolási és elérési módjától.

9. Logikai szintű adatfüggetlenség

Logikai adatfüggetlenség akkor áll fenn, ha az adatbázis logikai szerkezetének bővítése nem igényli az adatbázist használó alkalmazások (programok) megváltoztatását.

10. Jóság (integritás) függetlenség

Az adatok jóságának (érvényességének) korlátait az adatfeldolgozási programoktól függetlenül kell tudni meghatározni, és azokat katalógusban kell nyilvántartani. Legyen lehetséges a szóban forgó korlátokat megváltoztatni, anélkül hogy a meglévő alkalmazásokon változtatni kelljen.

11. Elosztástól való függetlenség

A meglévő alkalmazások működése zavartalan kell, hogy maradjon

(a) amikor sor kerül az adatbázis-kezelő osztott változatának bevezetésére

(b) amikor a meglévő osztott adatokat a rendszer újra szétosztja.

12. Megkerülhetetlenség szabálya

Ha a rendszernek van egy alacsony szintű (egyszerre egy rekordot érintő) interfésze, akkor ezt az interfészt ne lehessen a rendszer megkerülésére használni, például a relációs biztonsági vagy jósági (integritás védelmi) korlátok megsértésével.

Adatbázisrendszerek

ABR1 1. fejezet (19.- 45. oldal)

- **Adatbázis-kezelés:**
  - **Háttértárolón tárolt, nagy adatmennyiség hatékony kezelése (lekérdezése, módosítása)**
  - Adatmodell támogatása
  - Adatbázis-kezelő nyelvek támogatása
  - Több felhasználó támogatása
  - Tranzakció-kezelés
  - Helyreállíthatóság
  - Ügyfél-kiszolgáló felépítés
  - Adatvédelem, adatbiztonság

Adatmodellek

- **Az adatmodell a valóság fogalmainak, kapcsolatainak, tevékenységeinek magasabb színtű ábrázolása**
  - Hálós, hierarchikus adatmodell (apa-fiú kapcsolatok gráfja, hatékony keresés)
  - Relációs adatmodell (táblák rendszere, könnyen megfogalmazható műveletek)
  - Objektum-orientált adatmodell (az adatbázis-kezelés funkcionalitásainak biztosítása érdekében gyakran relációs adatmodellre épül)
  - Logikai adatmodell (szakértői rendszerek, tények és következtetési szabályok rendszere)
  - Félig strukturált (XML) adatmodell

Adatbázis-kezelő nyelvek

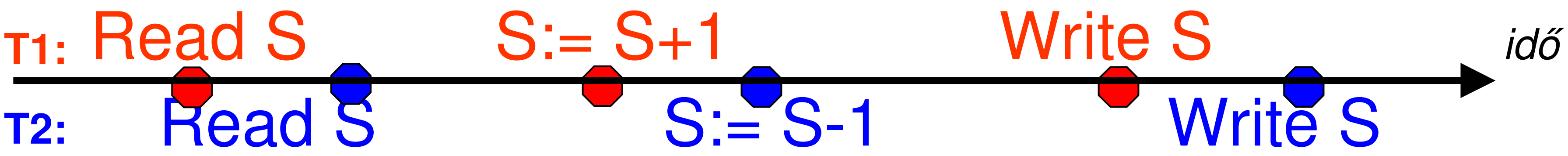
- **DDL** – adatdefiniáló nyelv (sémák, adatstruktúrák megadása)
- **DML** – adatkezelő nyelv (beszúrás, törlés, módosítás)
- **QL** – lekérdező nyelv
  - **Deklaratív** (SQL, kalkulusok)
  - **Procedurális** (relációs algebra)
- **PL/SQL** – programozási szerkezetek + SQL
- **Programozási nyelvbe ágyazás** (előfordító használata)
- **4GL** nyelvek (alkalmazások generálása)

Több felhasználó támogatása

- **Felhasználói csoportok**
- **DBA** – adatbázis-rendszergazda
- Jogosultságok (objektumok olvasása, írása, módosítása, készítése, törlése, jogok továbbadása, jogok visszavonása)
- Jogosultságok tárolása rendszertáblákban történik

Tranzakció-kezelés

- **Tranzakció**: adatkezelő műveletekből (adategység írása, olvasása) álló sorozat
- Cél: tranzakciók párhuzamos végrehajtása



- A tranzakció-kezelő biztosítja:
  - **Atomosság** (a tranzakció egységesen lefut vagy nem)
  - **Következetesség** (a tranzakció futása után konzisztens legyen az adatbázis)
  - **Elkülönítés** (párhuzamos végrehajtás eredménye egymás utáni végrehajtással egyezzen meg)
  - **Tartósság** (a befejezett tranzakció eredménye rendszerhiba esetén sem veszhet el)

Tranzakció-kezelés

- **Zárolások** (**Lock**, **Unlock**)

T1: (Lock S, Read S, S:=S+1, Write S, Unlock S)

T2: (Lock S, Read S, S:=S-1, Write S, Unlock S)

- A zár kiadásához meg kell várni a zár feloldását.
- Csökken a párhuzamosíthatóság
- Zárak finomsága (zárolt adategység nagysága, zárolás típusa) növeli a párhuzamosíthatóságot
- **Holtpont probléma:**

Lock A Read A Lock B Read B C:=A+B …….

Lock B Read B Lock A Read A D:=A+B ……..

T1 vár T2-re B miatt

T1 ⇄ T2

T2 vár T1-re A miatt

Tranzakció-kezelés

- **Kétfázisú protokoll** – a tranzakció elején zárolunk minden szükséges adatelemet, a végén minden zárat feloldunk
- **Tranzakciók érvényesítése**, naplózás, Commit, Rollback, Checkpoint
- **Ütemező** (tranzakciók műveleteinek végrehajtási sorrendjét adja meg)
- **Szérializálhatóság** (az ütemezés ekvivalens a tranzakciók egymás utáni végrehajtásával)
- Tranzakciók állapotát, elvégzett műveleteket rendszertáblák tárolják

Helyreállíthatóság

- Szoftver- vagy hardverhiba esetén az **utolsó konzisztens állapot visszaállítása**
- Rendszeres mentések
  - Statikus adatbázis (módosítás nem gyakori)
  - Dinamikus adatbázis (módosítás gyakori)
- Naplóállományok
- Összefügg a tranzakciókezeléssel

Ügyfél-kiszolgáló felépítés

- **Kiszolgáló:**
  - nagy tárhellyel rendelkező, gyors gép
  - adatbázis-műveletek optimalizált, párhuzamos végrehajtása
- **Ügyfél:**
  - adatbázis-művelet megfogalmazása
  - elküldése
  - az eredményadatok fogadása
  - megjelenítése
- Más felépítések is léteznek (például **köztes réteg** az ügyfél és a kiszolgáló között)

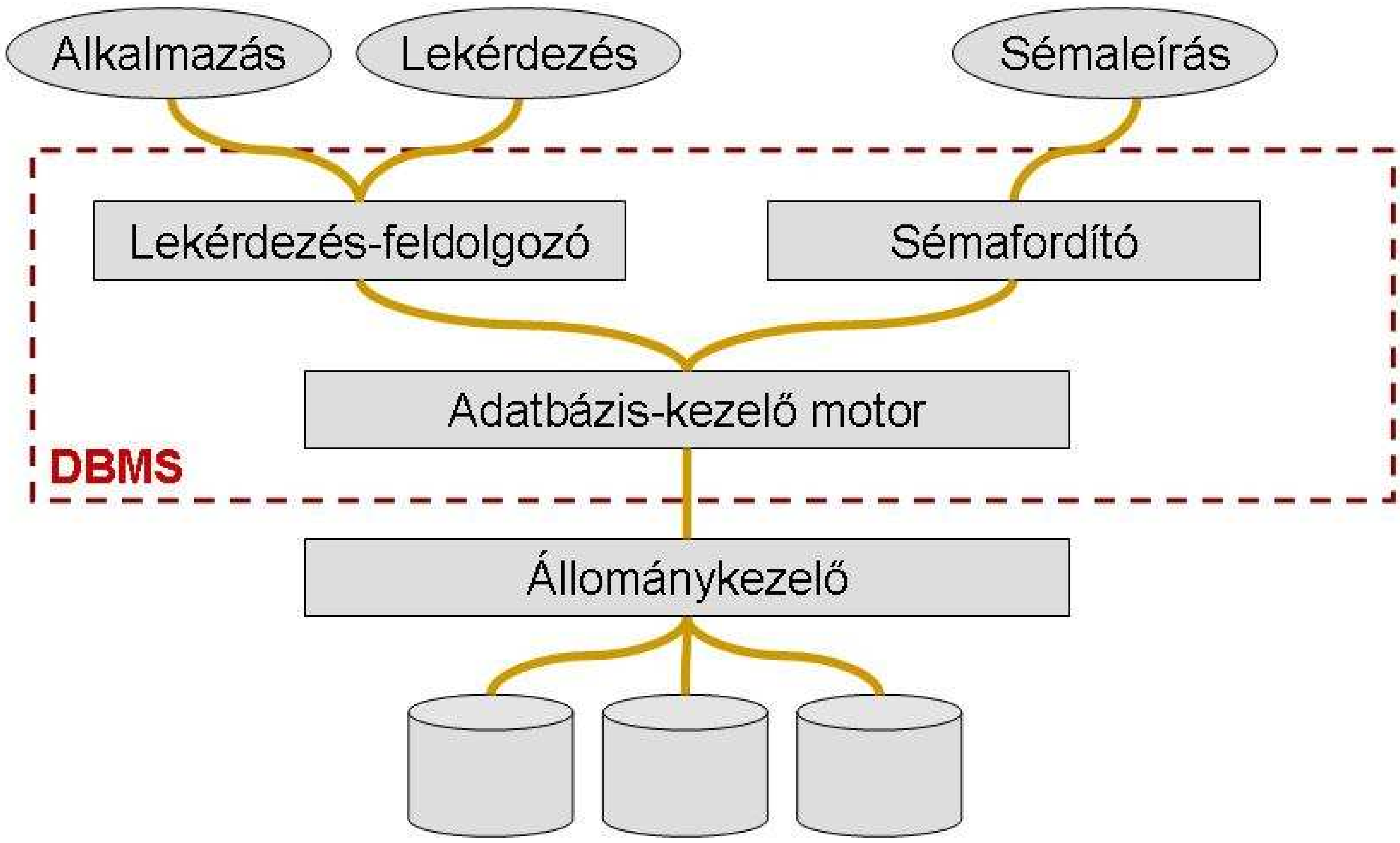
Adatvédelem, adatbiztonság

- **Jogosultságok kezelése**, felhasználók, jelszavak, hozzáférési jogok
- Adatbázissémák korlátozása (virtuális) **nézettáblák** segítségével
- Tárolt adatok, hálózati adatforgalmak **titkosítása** (nagy prímszámok, RSA, DES)

Adatbázis-kezelők felépítése

- **Lekérdezés-feldolgozó**
  - Lekérdezés szintaktikai ellenőrzése
  - Adatbázis-objektumok létezésének, és a hozzáférési jogoknak az ellenőrzése (metaadatbázis, rendszertáblák)
  - Lekérdezés optimális átfogalmazása
  - Végrehajtási tervek készítése
  - Az adatstruktúrák, méretek statisztikái alapján várhatóan minimális költségű végrehajtási terv kiválasztása
  - Az optimális végrehajtási terv lefuttatása
- **Tranzakció-kezelő:**
  - Tranzakciók párhuzamos végrehajtásának biztosítása (**a**tomosság, **k**övetkezetesség, **e**lkülönítés, **t**artósság)
- Tárkezelő (**állománykezelő**):
  - fizikai adatstruktúrák, táblák, **indexek**, **pufferek** kezelése

Adatbázisok ANSI/X3/SPARC modellje



Adatbázisok különböző szintjei

- **Sémák** (tervek, leírások) és **előfordulások** (konkrét adatok, megvalósulások)
- **Fizikai**, **logikai**, **alkalmazói réteg**:

<table>
<tr><th></th><th>Séma</th><th>Egy előfordulás</th></tr>
<tr><td>Alkalmazások</td><td>Select sum(fiz) as összfiz from Bér;</td><td>30</td></tr>
<tr><td>Logikai adatbázis</td><td>Bér(név, fiz)</td><td><table><tr><th>név</th><th>fiz</th></tr><tr><td>Kiss</td><td>10</td></tr><tr><td>Nagy</td><td>20</td></tr></table></td></tr>
<tr><td>Fizikai adatbázis</td><td>szekvenciális</td><td>(Bér,név,fiz,#2,Kiss,10,Nagy,20)</td></tr>
</table>

Adatbázisok különböző szintjei

- **Fizikai adatfüggetlenség**
  - Fizikai adatbázis módosítása (indexek készítése, az adatok más adatstruktúrákban tárolása) nem látszik a felette levő szinteken
  - Hatékonyság növelhető jobb tárolási struktúrákkal
- **Logikai adatfüggetlenség**
  - A logikai adatbázis **bővítése** (új táblák, oszlopok hozzáadása) esetén a régi alkalmazások változtatás nélkül ugyanúgy működjenek

Relációs adatmodell

ABR1 3. fejezet (104.- 110. oldal)
ABR1 4. fejezet (196.- 215. oldal)

- **Relációséma**: R(A1,A2,…,An)
  - R – relációnév
  - Ai – attribútum- vagy tulajdonságnevek, oszlopnevek
  - Dom(Ai) – lehetséges értékek halmaza, típusa
  - Egy sémán belül az attribútumok különbözőek
- **Reláció-előfordulás**: r
  - r - reláció, tábla, sorhalmaz
  - Egy sor egyszer szerepel
  - Sorok sorrendje lényegtelen
  - Oszlopok sorrendje lényegtelen

$$r \subseteq \mathop{\mathrm{X}}_{i=1}^{n} \mathrm{Dom}(Ai)$$

Relációs adatmodell

- **Jelölések**
  - $t \in r$ esetén t sor (angolul: tuple – n-es)
    - t(Ai) vagy t($i) – a t sor i-edik komponense
    - t[Ai1,...,Aik] - a t sor i1,…,ik-adik komponenseiből álló vektor
- Különböző sémák azonos attribútumai esetén
  - R.A – prefixszel különböztetjük meg
- Egy t sor függvénynek is tekinthető

$t:\{A1,\ldots,An\} \to \bigcup_{i=1}^{n} \mathrm{Dom}\,(\mathrm{Ai})$ ahol $t(Ai) \in Dom(Ai)$, i=1..n

Példa

Bér

név	fiz	kor	
Kiss	10	35	t1
Nagy	20	45	t2
Kovács	15	22	t3

t1(név)=„Kiss”

t3($3)=22

t2(név,kor)=(„Nagy”,45)

t1(Bér.fiz)=10

Relációs algebra

- **Relációs algebra = kifejezések halmaza**
- **Kifejezés**:
  - konstans reláció
  - relációs változó
  - alapoperátorok véges sok alkalmazása kifejezésekre
  - ezek és csak ezek
- **Alapoperátorok:**
  1. **Egyesítés**
  2. **Különbség**
  3. **Descartes-szorzat**
  4. **Vetítés**
  5. **Kiválasztás**
  6. **Átnevezés**

Egyesítés, unió

- r, s és $r \cup s$ azonos sémájú
- **$r \cup s$** $:= \{t \mid t \in r \text{ vagy } t \in s\}$
- $|r \cup s| <= |r|+|s|$, ahol $|r|$ az r reláció sorainak száma
- azonos sor csak egyszer szerepelhet

A	B
0	0
0	1

$\cup$

A	B
0	0
1	0

=

A	B
0	0
0	1
1	0

Kivonás, különbség

- r, s és r - s azonos sémájú
- **r - s** := { t | t ∈ r és t ∉ s}
- |r - s| <= |r|

A	B
0	0
0	1

–

A	B
0	0
1	0

=

A	B
0	1

Szorzás, direktszorzat vagy Descartes-szorzat

- r, s sémáiban nincs közös attribútum
- $r \times s$ sémája a sémák egyesítése
- $\mathbf{r} \times \mathbf{s} := \{ t \mid t[R] \in r \text{ és } t[S] \in s \}$
- $|r \times s| = |r| * |s|$

A	B
0	0
0	1

$\times$

C	D
0	0
1	0

=

A	B	C	D
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0

Vetítés, projekció

- $X \subseteq \{A1,\ldots,An\}$
- $\Pi_X(r)$ sémája X
- **$\Pi_X(r)$** := { t | van olyan $t' \in r$, melyre $t'[X] = t$ }
- $|\Pi_X(r)| \le |r|$

r:

A	B	C	D
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0

$\Pi_{BD}(r) =$

B	D
0	0
1	0

$\Pi_{DA}(r) =$

D	A
0	0

Kiválasztás, szűrés, szelekció

- $\sigma_F(r)$ és r sémája megegyezik
- $\boldsymbol{\sigma_F(r)}$:= { t | t∈r és F(t) = IGAZ }
- **F feltétel:**
  - atomi, elemi feltétel
    - **Ai Θ Aj**, ahol Θ ∈ { **=, ≠, <,>, <=, >=** }
    - **Ai Θ c**, **c Θ Ai** ahol c egy konstans
  - feltételekből **∧, ∨, ¬** logikai összekapcsolókkal, és zárójelekkel kapható kifejezés

r:

A	B	C	D
0	0	0	0
0	0	1	0
0	1	0	0

$\sigma_{\mathbf{A=C} \wedge \neg \mathbf{(B<1)}}(r) =$

A	B	C	D
0	1	0	0

Kiválasztás, szűrés, szelekció

- $|\sigma_F(r)| <= |r|$
- a feltételben függvények nem használhatók:

  $\boldsymbol{\sigma_{A + B < 5}(r)}$ nem megengedett
- **az összetett feltételek átírhatók elemi feltételeket használó kifejezésekké a következő szabályok segítségével**:
  - $\boldsymbol{\sigma_{F1 \wedge F2}(r) \cong \sigma_{F1}(\sigma_{F2}(r)) \cong \sigma_{F2}(\sigma_{F1}(r))}$
  - $\boldsymbol{\sigma_{F1 \vee F2}(r) \cong \sigma_{F1}(r) \cup \sigma_{F2}(r)}$
  - A De Morgan azonosság segítségével a negáció beljebb vihető:
    - $\boldsymbol{\neg (F1 \wedge F2)}$ helyett $\boldsymbol{(\neg F1) \vee (\neg F2)}$
    - $\boldsymbol{\neg (F1 \vee F2)}$ helyett $\boldsymbol{(\neg F1) \wedge (\neg F2)}$
    - elemi feltétel tagadása helyett a fordított összehasonlítást használjuk:

      például $\neg$ **(A < B)** helyett **(A >= B)**

Kiválasztás, szűrés, szelekció

$$\sigma_{(\neg(A = C \wedge \neg (B < 1))) \wedge (D < 2)}(r) =$$

$$\sigma_{(\neg(A = C) \vee \neg\neg(B < 1)) \wedge (D < 2)}(r) =$$

$$\sigma_{A \neq C}(\sigma_{D < 2}(r)) \cup \sigma_{B < 1}(\sigma_{D < 2}(r))$$

- az elemi feltételekhez lekérdezést gyorsító adatszerkezetek, indexek készíthetők

Átnevezés

- A relációnak és az attribútumoknak új nevet adhatunk.
- Ha r sémája R(A1,…,An), akkor $\rho_{S(B1,\ldots,Bn)}(r)$ sémája **S(B1,…,Bn).**
- $|\rho_{S(B1,\ldots,Bn)}(r)| = |r|$

$\rho_{MUNKA(dolg,jöv)}(r) =$

BÉR

r:

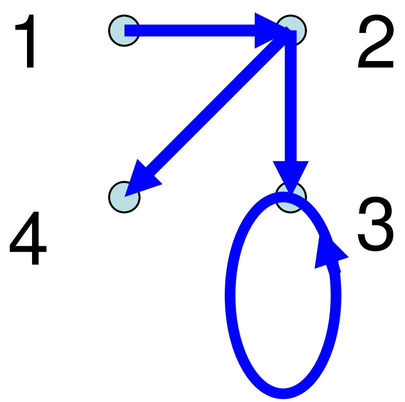
név	fiz
Kiss	10
Nagy	20

MUNKA

dolg	jöv
Kiss	10
Nagy	20

Tranzitív lezárás

- ÉL(honnan, hova)
- ÚT(honnan, hova) – tranzitív lezárás



ÉL

honnan	hova
1	2
2	4
2	3
3	3

ÚT

honnan	hova
1	2
2	4
2	3
3	3
1	3
1	4

- nem triviális rekurzió
- TÉTEL: Nem létezik olyan relációs algebrai kifejezés, amelyet tetszőleges ÉL táblára alkalmazva a neki megfelelő ÚT táblát eredményezi.

Származtatott műveletek

- A gyakran használt kifejezések helyett új műveleteket vezetünk be.
- Nem alapműveletek, hanem származtatottak
- **Metszet**
  - **$r \cap s = \{ t \mid t \in r$ és $t \in s \}$**
  - többféleképpen kifejezhető relációs algebrában:
    - **$r \cap s = r - (r - s) = s - (s - r) = r \cup s - ((r - s) \cup (s - r))$**
- **Összekapcsolások (JOIN)**
  - Téta-összekapcsolás (**Θ-join**)
  - Egyen-összekapcsolás (**equi-join**)
  - Természetes összekapcsolás (**natural join**)
  - Félig-összekapcsolás (**semi-join**)
  - Külső összekapcsolás (**outer join**)
- A szorzáshoz hasonlóan költséges műveletek, nagy méretű táblákat eredményezhetnek, kivételt képez a félig-összekapcsolás.

Téta-összekapcsolás

- r, s sémáiban (R(A1,…,An), S(B1,…,Bn) nincs közös attribútum
- $r \underset{Ai\Theta Bj}{\bowtie} s = \sigma_{Ai\ \Theta\ Bj}(r \times s)$

A	B
0	0
0	1

$\underset{B=C}{\bowtie}$

C	D
0	0
0	1

=

A	B	C	D
0	0	0	0
0	0	0	1

- **Ai=Bj** feltétel esetén **egyen-összekapcsolás**nak hívjuk.

Természetes összekapcsolás

- r, s sémái R(A1,…,An,B1,…,Bk), illetve S(B1,…,Bk,C1,…,Cm)
- r $\bowtie$ s =

$$\rho_{P(A1,\dots,An,B1,\dots,Bk,C1,\dots,Cm)}\Pi_{A1,\dots,An,R.B1,\dots,R.Bk,C1,\dots,Cm}\sigma_{R.B1=S.B1\wedge\dots\wedge R.Bk=S.Bk}(r\times s)$$

A	B
0	0
2	1
1	2

$\bowtie$

B	C
0	0
0	2
1	3
4	3

=

A	B	C
0	0	0
0	0	2
2	1	3

Félig-összekapcsolás

- r, s sémái R(A1,…,An,B1,…,Bk), illetve S(B1,…,Bk,C1,…,Cm)
- $r \ltimes s = \rho_{P(A1,\ldots,An,B1,\ldots,Bk,)} \Pi_{A1,\ldots,An,R.B1,\ldots,R.Bk} (r|\times|s)$
- Az első relációban mely sorokhoz létezik kapcsolható sor a második táblából

A	B
0	0
2	1
1	2

$\ltimes$

B	C
0	0
0	2
1	3
4	3

=

A	B
0	0
2	1

Külső összekapcsolás

- Nem relációs algebrai művelet, mert kilép a modellből
- r, s sémái R(A1,…,An,B1,…,Bk), illetve S(B1,…,Bk,C1,…,Cm)
- r $\overset{o}{\bowtie}$ s = r $\bowtie$ s relációt kiegészítjük az r és s soraival, a hiányzó helyekre NULL értéket írva

A	B
0	0
2	1
1	2

$\overset{o}{\bowtie}$

B	C
0	0
0	2
1	3
4	3

=

A	B	C
0	0	0
0	0	2
2	1	3
1	2	NULL
NULL	4	3

Összekapcsolások

- Ha r, s **sémái megegyeznek**, akkor $\mathbf{r|\times|s = r \cap s}$.
- Ha r, s sémáiban **nincs közös attribútum**, akkor $\mathbf{r|\times|s = r\times s}$.
- Ha $r = \emptyset$, akkor $r\times s = \emptyset$ és $r|\times|s = \emptyset$.
- A külső összekapcsolás lehet bal oldali, ha csak r sorait vesszük hozzá a természetes összekapcsolás-hoz: $\mathbf{r|\overset{o}{\times}|_B s}$. Hasonlóan értelmezhetjük a jobb oldali összekapcsolást is $\mathbf{r|\overset{o}{\times}|_J s}$.

Osztás, hányados

- Maradékos osztás: $7 \div 3 = 2$, mert 2 a legnagyobb egész, amelyre még $2 * 3 \leq 7$.
- Relációk szorzata esetén $\leq$ helyett tartalmazás.
- r és s sémája **R**(**A1,…,An**,**B1,…,Bm**), illetve **S**(B1,…,Bm), **r $\div$ s** sémája R(**A1,…,An**)
- **r $\div$ s** a legnagyobb (**legtöbb sort tartalmazó**) reláció, amelyre **(r $\div$ s) $\times$ s $\subseteq$ r**.
- Kifejezhető relációs algebrában:
- $\Pi_{A1,\ldots,An}(r) - \Pi_{A1,\ldots,An}((\Pi_{A1,\ldots,An}(r) \times s - r)$
- Lehetséges értékekből kivonjuk a rossz értékeket.
- $(p \times r) \div r = p$

Osztás, hányados

- Ki szereti legalább azokat, mint Micimackó?

KI	MIT
Füles	málna
Füles	méz
Füles	alma
Micimackó	málna
Micimackó	méz
Kanga	málna
Kanga	körte
Nyuszi	lekvár

÷

MIT
málna
méz

=

KI
Füles
Micimackó

$$\text{szeret} \div \Pi_{MIT}(\sigma_{KI='Micimackó'}(\text{szeret}))$$

Monotonitás

- **Monoton nem csökkenő** (röviden **monoton**) kifejezés: bővebb relációra alkalmazva az eredmény is bővebb:

Ha $\mathbf{R_i} \subseteq \mathbf{S_i}$, i=1,…,n, akkor $\mathbf{E(R_1,\ldots,R_n)} \subseteq \mathbf{E(S_1,\ldots,S_n)}$.

- A kivonás kivétel az alapműveletek monoton műveletek (monoton relációs algebra).

A	B
0	1
0	0

\-

A	B
0	1

$\not\subseteq$

A	B
0	1
0	0

\-

A	B
0	1
0	0

Monotonitás

- **DE**: Monoton kifejezésben is szerepelhet kivonás: $\mathbf{r} \cap \mathbf{s} = r - (r - s)$ monoton.
- Ha E, E1, Ek monoton kifejezések, és E(E1(…),…,Ek(…)) helyes kifejezés, akkor monoton is.
- **Következmény**: **A kivonás nem fejezhető ki a többi alapművelettel**.

FELADATOK

- **Relációs algebrai alapműveleteket ($\cup$, -, $\times$, Π, σ, ρ) tartalmazó kifejezésekkel fejezzük ki a következő lekérdezéseket!**
- **Legyen a relációséma: szeret(név,gyümölcs), röviden s(n,g).**

név	gyümölcs
Füles	málna
Füles	körte
Füles	alma
Micimackó	málna
Micimackó	körte
Kanga	málna
Kanga	körte
Nyuszi	eper

1. Milyen gyümölcsöket szeret Micimackó?

1. Megoldás:

$$m1 := \Pi_{g}(\sigma_{n='Micimackó'}(s))$$

FELADATOK

- **Relációs algebrai alapműveleteket ($\cup$, -, $\times$, Π, σ, ρ) tartalmazó kifejezésekkel fejezzük ki a következő lekérdezéseket!**
- **Legyen a relációséma: szeret(név,gyümölcs), röviden s(n,g).**

név	gyümölcs
Füles	málna
Füles	körte
Füles	alma
Micimackó	málna
Micimackó	körte
Kanga	málna
Kanga	körte
Nyuszi	eper

2. Melyek azok a gyümölcsök, amelyeket Micimackó NEM szeret (de valaki más igen)?

2. Megoldás:

$$m1 := \Pi_g(\sigma_{n='Micimackó'}(s))$$

$$gy := \Pi_g(s)$$

$$m2 := gy - m1$$

FELADATOK

- **Relációs algebrai alapműveleteket ($\cup$, -, $\times$, $\prod$, σ, ρ) tartalmazó kifejezésekkel fejezzük ki a következő lekérdezéseket!**
- **Legyen a relációséma: szeret(név,gyümölcs), röviden s(n,g).**

név	gyümölcs
Füles	málna
Füles	körte
Füles	alma
Micimackó	málna
Micimackó	körte
Kanga	málna
Kanga	körte
Nyuszi	eper

3. Kik szeretik az almát?

3. Megoldás:

$$m3 := \prod_{n}(\sigma_{g='alma'}(s))$$

FELADATOK

- **Relációs algebrai alapműveleteket** **($\cup$, -, $\times$, Π, σ, ρ)** **tartalmazó kifejezésekkel fejezzük ki a következő lekérdezéseket!**
- **Legyen a relációséma: szeret(név,gyümölcs), röviden s(n,g).**

név	gyümölcs
Füles	málna
Füles	körte
Füles	alma
Micimackó	málna
Micimackó	körte
Kanga	málna
Kanga	körte
Nyuszi	eper

4. Kik NEM szeretik az almát, de valami mást szeretnek?

4. Megoldás:

$$m3 := \Pi_n(\sigma_{g='alma'}(s))$$

$$k := \Pi_n(s)$$

$$m4 := k - m3$$

ROSSZ MEGOLDÁS:

$$\Pi_n(\sigma_{g\neq'alma'}(s))$$

Füles szeret olyat, ami nem az alma!

FELADATOK

- **Relációs algebrai alapműveleteket ($\cup$, -, $\times$, $\prod$, σ, ρ) tartalmazó kifejezésekkel fejezzük ki a következő lekérdezéseket!**
- **Legyen a relációséma: szeret(név,gyümölcs), röviden s(n,g).**

név	gyümölcs
Füles	málna
Füles	körte
Füles	alma
Micimackó	málna
Micimackó	körte
Kanga	málna
Kanga	körte
Nyuszi	eper

5. Kik szeretnek almát VAGY körtét?

5. Megoldás:

$$m3 := \prod_{n}(\sigma_{g='alma'}(s))$$

$$m31 := \prod_{n}(\sigma_{g='körte'}(s))$$

$$m5 := m3 \cup m31$$

FELADATOK

- **Relációs algebrai alapműveleteket ($\cup$, -, $\times$, $\prod$, σ, ρ) tartalmazó kifejezésekkel fejezzük ki a következő lekérdezéseket!**
- **Legyen a relációséma: szeret(név,gyümölcs), röviden s(n,g).**

név	gyümölcs
Füles	málna
Füles	körte
Füles	alma
Micimackó	málna
Micimackó	körte
Kanga	málna
Kanga	körte
Nyuszi	eper

6. Kik szeretnek almát ÉS körtét?

6. Megoldás:

$$m3 := \prod_{n}(\sigma_{g='alma'}(s))$$

$$m31 := \prod_{n}(\sigma_{g='körte'}(s))$$

$$m6 := m3 \cap m31 = m3 - (m3 - m31)$$

FELADATOK

- **Relációs algebrai alapműveleteket** **($\cup$, -, $\times$, Π, σ, ρ) tartalmazó kifejezésekkel fejezzük ki a következő lekérdezéseket!**
- **Legyen a relációséma: szeret(név,gyümölcs), röviden s(n,g).**

név	gyümölcs
Füles	málna
Füles	körte
Füles	alma
Micimackó	málna
Micimackó	körte
Kanga	málna
Kanga	körte
Nyuszi	eper

7. Kik szeretik a körtét, de az almát NEM?

7. Megoldás:

$$m3 := \Pi_n(\sigma_{g='alma'}(s))$$
$$m31 := \Pi_n(\sigma_{g='körte'}(s))$$
$$m7 := m31 - m3$$

FELADATOK

- **Relációs algebrai alapműveleteket (∪, -, ×, Π, σ, ρ) tartalmazó kifejezésekkel fejezzük ki a következő lekérdezéseket!**
- **Legyen a relációséma: szeret(név,gyümölcs), röviden s(n,g).**

név	gyümölcs
Füles	málna
Füles	körte
Füles	alma
Micimackó	málna
Micimackó	körte
Kanga	málna
Kanga	körte
Nyuszi	eper

8. Kik szeretnek legalább kétféle gyümölcsöt?

8. Megoldás:

Próbáljuk a d:= s1 × s2 szorzatot felhasználni!

Aki több gyümölcsöt is szeret, ahhoz több sor fog tartozni a szorzatban.

FELADATOK

8. Megoldás: $m8 := \prod_{s1.n}(\sigma_{s1.n=s2.n \wedge s1.g \neq s2.g}(s1 \times s2))$

$s1.n = s2.n$

$s1.g \neq s2.g$

s1.n	s1.g
Füles	málna
Füles	körte
Füles	alma
Micimackó	málna
Micimackó	körte
Kanga	málna
Kanga	körte
Nyuszi	eper

$\times$

s2.n	s2.g
Füles	málna
Füles	körte
Füles	alma
Micimackó	málna
Micimackó	körte
Kanga	málna
Kanga	körte
Nyuszi	eper

FELADATOK

név	gyümölcs
Füles	málna
Füles	körte
Füles	alma
Micimackó	málna
Micimackó	körte
Kanga	málna
Kanga	körte
Nyuszi	eper

9. Kik szeretnek legalább HÁROMFÉLE gyümölcsöt?

9. Megoldás:

Próbáljuk a d:= s1 $\times$ s2 $\times$ s3 szorzatot felhasználni!

m9 :=

$$\Pi_{s1.n}(\sigma_{s1.n=s2.n \wedge s1.n=s3.n \wedge s1.g \neq s2.g \wedge s1.g \neq s3.g \wedge s2.g \neq s3.g}(s1 \times s2 \times s3))$$

FELADATOK

- **Relációs algebrai alapműveleteket ($\cup$, -, $\times$, $\prod$, σ, ρ) tartalmazó kifejezésekkel fejezzük ki a következő lekérdezéseket!**
- **Legyen a relációséma: szeret(név,gyümölcs), röviden s(n,g).**

név	gyümölcs
Füles	málna
Füles	körte
Füles	alma
Micimackó	málna
Micimackó	körte
Kanga	málna
Kanga	körte
Nyuszi	eper

10. Kik szeretnek legfeljebb kétféle gyümölcsöt (1 vagy 2 gyümölcsöt)?

10. Megoldás:

Akik legalább háromfélét szeretnek, azok pont nem ilyenek!

$$k := \prod_n(s)$$

$$m10 := k - m9$$

FELADATOK

- **Relációs algebrai alapműveleteket ($\cup$, -, $\times$, Π, σ, ρ) tartalmazó kifejezésekkel fejezzük ki a következő lekérdezéseket!**
- **Legyen a relációséma: szeret(név,gyümölcs), röviden s(n,g).**

név	gyümölcs
Füles	málna
Füles	körte
Füles	alma
Micimackó	málna
Micimackó	körte
Kanga	málna
Kanga	körte
Nyuszi	eper

11. Kik szeretnek pontosan kétféle gyümölcsöt?

11. Megoldás:

Akik legalább kétfélét szeretnek, és ugyanakkor legfeljebb kétfélét szeretnek, azok pontosan kétfélét szeretnek.

$$m11 := m8 \cap m10 =$$

$$m8 - (m8 - m10)$$

FELADATOK

- **Relációs algebrai alapműveleteket ($\cup$, -, $\times$, Π, σ, ρ) tartalmazó kifejezésekkel fejezzük ki a következő lekérdezéseket!**
- **Legyen a relációséma: szeret(név,gyümölcs), röviden s(n,g).**

név	gyümölcs
Füles	málna
Füles	körte
Füles	alma
Micimackó	málna
Micimackó	körte
Kanga	málna
Kanga	körte
Nyuszi	alma

12. Kik szeretik az összes olyan gyümölcsöt, amit valaki szeret?

12. Megoldás:

Az összes gyümölcsnek a név mellett kellene látszani: OSZTÁS!

$$gy := \Pi_g(s)$$

$$m12 := s \div gy$$

FELADATOK

- **Relációs algebrai alapműveleteket (∪, -, ×, Π, σ, ρ) tartalmazó kifejezésekkel fejezzük ki a következő lekérdezéseket!**
- **Legyen a relációséma: szeret(név,gyümölcs), röviden s(n,g).**

név	gyümölcs
Füles	málna
Füles	körte
Füles	alma
Micimackó	málna
Micimackó	körte
Kanga	málna
Kanga	körte
Nyuszi	eper

13. Kik szeretik az összes olyan gyümölcsöt, amit Micimackó szeret (esetleg mást is szerethetnek)?

13. Megoldás:

Az összes Micimackó által kedvelt gyümölcsnek a név mellett kellene látszani: OSZTÁS!

$$m1:=\Pi_{g}(\sigma_{n='Micimackó'}(s))$$

$$m13:= s \div m1$$

FELADATOK

név	gyümölcs
Füles	málna
Füles	körte
Füles	alma
Micimackó	málna
Micimackó	körte
Kanga	málna
Kanga	körte
Nyuszi	körte

14. Kik szeretnek legfeljebb olyan gyümölcsöket, amiket Micimackó is szeret (azaz olyat nem szeretnek, amit Micimackó sem)?

14. Megoldás:

Készítsünk egy táblát, hogy ki miket nem szeret:

$$ns := \Pi_n(s) \times \Pi_g(s) - s$$

Azok kellenek, akik neve mellett az összes Micimackó által NEM kedvelt gyümölcs (m2) szerepel, esetleg még más gyümölcsök is: OSZTÁS!

$$m14 := ns \div m2$$

FELADATOK

- **Relációs algebrai alapműveleteket ($\cup$, -, $\times$, Π, σ, ρ) tartalmazó kifejezésekkel fejezzük ki a következő lekérdezéseket!**
- **Legyen a relációséma: szeret(név,gyümölcs), röviden s(n,g).**

név	gyümölcs
Füles	málna
Füles	körte
Füles	alma
Micimackó	málna
Micimackó	körte
Kanga	málna
Kanga	körte
Nyuszi	eper

15. Kik szeretik pontosan azokat a gyümölcsöket, amiket Micimackó szeret?

15. Megoldás:

Pontosan = legalább és legfeljebb!

$$m15:=m13 \cap m14= m13 - (m13 - m14)$$

FELADATOK

név	gyümölcs
Füles	málna
Füles	körte
Füles	alma
Micimackó	málna
Micimackó	körte
Kanga	málna
Kanga	körte
Nyuszi	eper

16. Melyek azok a (név,név) párok, akiknek legalább egy gyümölcsben eltér az ízlésük, azaz az egyik szereti ezt a gyümölcsöt, a másik meg nem?

16. Megoldás:

Vegyük a d:= s1 $\times$ s2 szorzatot. Cseréljük fel a 2. és 4. oszlopot és hasonlítsuk össze a két táblát.

d1 :=$\prod_{\$1,\$4,\$3,\$2}(d)$

Ha n1 szereti g1-et, de n2 nem szeret g1-et, hanem g2-t, akkor

(n1,g1,n2,g2)$\in$ d, (n1,g2,n2,g1)$\in$ d1

viszont (n1,g2,n2,g1)$\notin$ d. Így

m16:= $\prod_{\$1,\$3}(d1-d)$

FELADATOK

név	gyümölcs
Füles	málna
Füles	körte
Füles	alma
Micimackó	málna
Micimackó	körte
Kanga	málna
Kanga	körte
Nyuszi	eper

17. Melyek azok a (név,név) párok, akiknek pontosan ugyanaz az ízlésük, azaz pontosan ugyanazokat a gyümölcsöket szeretik?

17. Megoldás:

Előző feladatban a komplementer párokat határoztuk meg.

$$nn := \Pi_{s1.n}(s1) \times \Pi_{s2.n}(s2)$$

$$m17 := nn - m16$$

FELADATOK

- **Relációs algebrai alapműveleteket ($\cup$, -, $\times$, Π, σ, ρ) tartalmazó kifejezésekkel fejezzük ki a következő lekérdezéseket!**
- **Legyen a relációséma: mézevők(név,csupor_szám), röviden me(n,c).**

név	csupor_szám
Füles	1
Micimackó	6
Kanga	3
Nyuszi	6

18. Kiknek van a legtöbb csupor mézük?

18. Megoldás:

A maximum az összes többi értéknél nagyobb vagy egyenlő. Képezzünk téta-összekapcsolást!

$$t := \sigma_{m1.c>=m2.c}(me1 \times me2)$$

Ha (n1,c1) maximális, akkor az összes (n2,c2) pár, azaz me2 megjelenik mellette a szorzatban: OSZTÁS!

$$m18 := \Pi_{m1.n}(t \div me2)$$

Hasonlóan a minimum is kifejezhető!

Lekérdezések optimalizálása

CÉL: A lekérdezéseket gyorsabbá akarjuk tenni a táblákra vonatkozó paraméterek, statisztikák, indexek ismeretében és általános érvényű tulajdonságok, heurisztikák segítségével.

Például, hogyan, milyen procedúrával értékeljük ki az alábbi SQL (deklaratív) lekérdezést?

Legyen adott R(A,B,C) és S(C,D,E). Melyek azok az R.B és S.D értékek azokban az R, illetve S táblabeli sorokban, amely sorokban R.A='c' és S.E=2 és R.C=S.C?

Ugyanez SQL-ben:

```
Select B,D
From R,S
Where R.A = 'c' and S.E = 2 and R.C=S.C;
```

Lekérdezések optimalizálása

R

A	B	C
a	1	10
b	1	20
c	2	10
d	2	35
e	3	45

S

C	D	E
10	x	2
20	y	2
30	z	2
40	x	1
50	y	3

A lekérdezés eredménye:

B	D
2	x

Lekérdezések optimalizálása

Hogy számoljuk ki tetszőleges tábla esetén az eredményt?

Egy lehetséges terv

- Vegyük a két tábla szorzatát!
- Válasszuk ki a megfelelő sorokat!
- Hajtsuk végre a vetítést!

- Ez a direktszorzaton alapuló összekapcsolás.
- Oracleben: NESTED LOOP.
- Nagyon költséges!

Lekérdezések optimalizálása

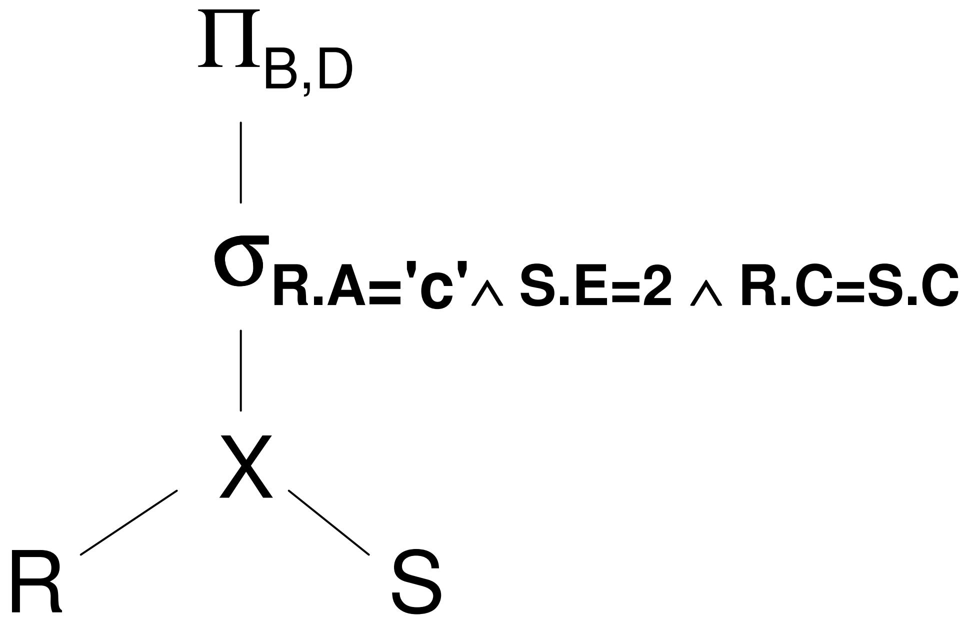
RXS

R.A	R.B	R.C	S.C	S.D	S.E
a	1	10	10	x	2
a	1	10	20	y	2
.					
.					
c	2	10	10	x	2
.					
.					

Ez a sor kell!

Lekérdezések optimalizálása

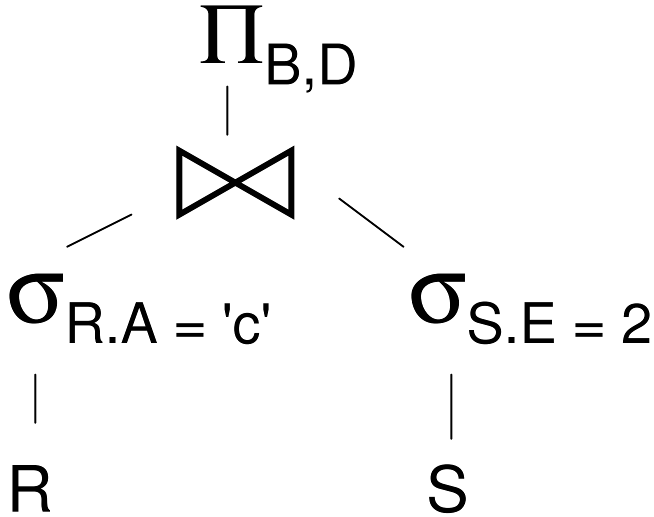
Ugyanez a terv relációs algebrában:



$$\Pi_{B,D} [\, \sigma_{R.A='c' \wedge S.E=2 \wedge R.C = S.C} (RXS)]$$

Lekérdezések optimalizálása

Egy másik lehetséges kiszámítási javaslat:



Lekérdezések optimalizálása

R

A	B	C
a	1	10
b	1	20
c	2	10
d	2	35
e	3	45

σ (R)

A	B	C
c	2	10

σ(S)

C	D	E
10	x	2
20	y	2
30	z	2

S

C	D	E
10	x	2
20	y	2
30	z	2
40	x	1
50	y	3

$\bowtie$

$\Pi_{B,D}$

Ugyanazt számolja ki!

B	D
2	x

Lekérdezések optimalizálása

Használjuk ki az R.A és S.C oszlopokra készített **indexeket**:

(1) Az R.A index alapján keressük meg az R azon sorait, amelyekre R.A = 'c'!

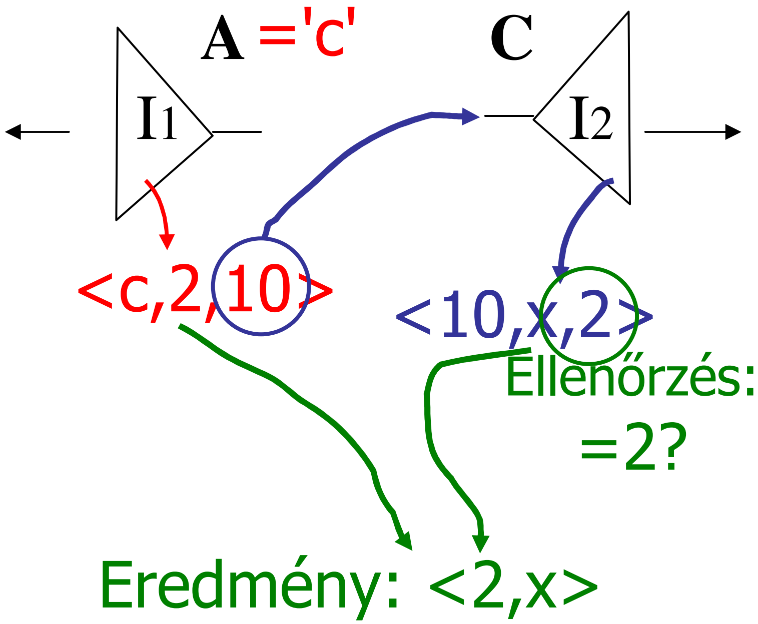
(2) Minden megtalált R.C értékhez az S.C index alapján keressük meg az S-ből az ilyen értékű sorokat!

(3) Válasszuk ki a kapott S-beli sorok közül azokat, amelyekre S.E = 2!

(4) Kapcsoljuk össze az R és S így kapott sorait, és végül vetítsünk a B és D oszlopokra.

R

A	B	C
a	1	10
b	1	20
c	2	10
d	2	35
e	3	45

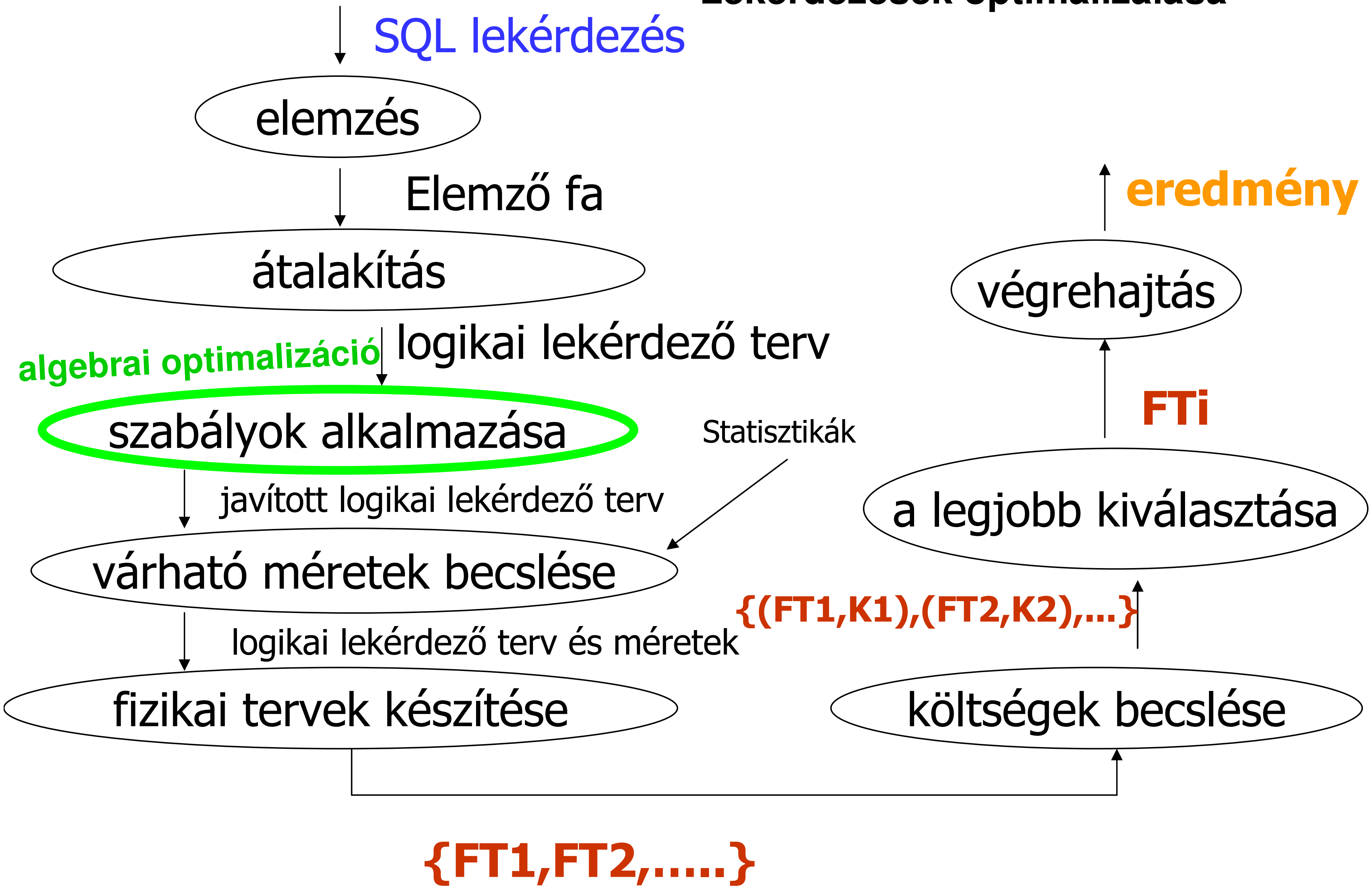


S

C	D	E
10	x	2
20	y	2
30	z	2
40	x	1
50	y	3

2-INDEXES ÖSSZEKAPCSOLÁS

Lekérdezések optimalizálása



Algebrai optimalizáció

- **Cél:** a relációs algebrai kifejezéseket minél gyorsabban akarjuk kiszámolni.
- **Költségmodell:** a kiszámítás költsége arányos a relációs algebrai kifejezés részkifejezéseinek megfelelő relációk tárolási méreteinek összegével.
- **Módszer:** a műveleti tulajdonságokon alapuló ekvivalens átalakításokat alkalmazunk, hogy várhatóan kisebb méretű relációk keletkezzenek.
- **Az eljárás heurisztikus**, tehát nem az argumentum relációk valódi méretével számol.
- **Az eredmény nem egyértelmű:** Az átalakítások sorrendje nem determinisztikus, így más sorrendben végrehajtva az átalakításokat más végeredményt kaphatunk, de mindegyik általában jobb költségű, mint amiből kiindultunk.
- **Megjegyzés:** Mivel az SQL bővebb, mint a relációs algebra, ezért az optimalizálást bővített relációs algebrára is meg kell adni, de először a hagyományos algebrai kifejezéseket vizsgáljuk.

Algebrai optimalizáció

- A relációs algebrai kifejezést **gráffal** ábrázoljuk.
- **Kifejezésfa:**
  - a **nem levél csúcsok**: a relációs algebrai műveletek:
    - unáris (σ,Π,ρ) – egy gyereke van
    - bináris ($-,\cup,\times$) – két gyereke van (bal oldali az első, jobb oldali a második argumentumnak felel meg)
  - a **levél csúcsok**: konstans relációk vagy relációs változók

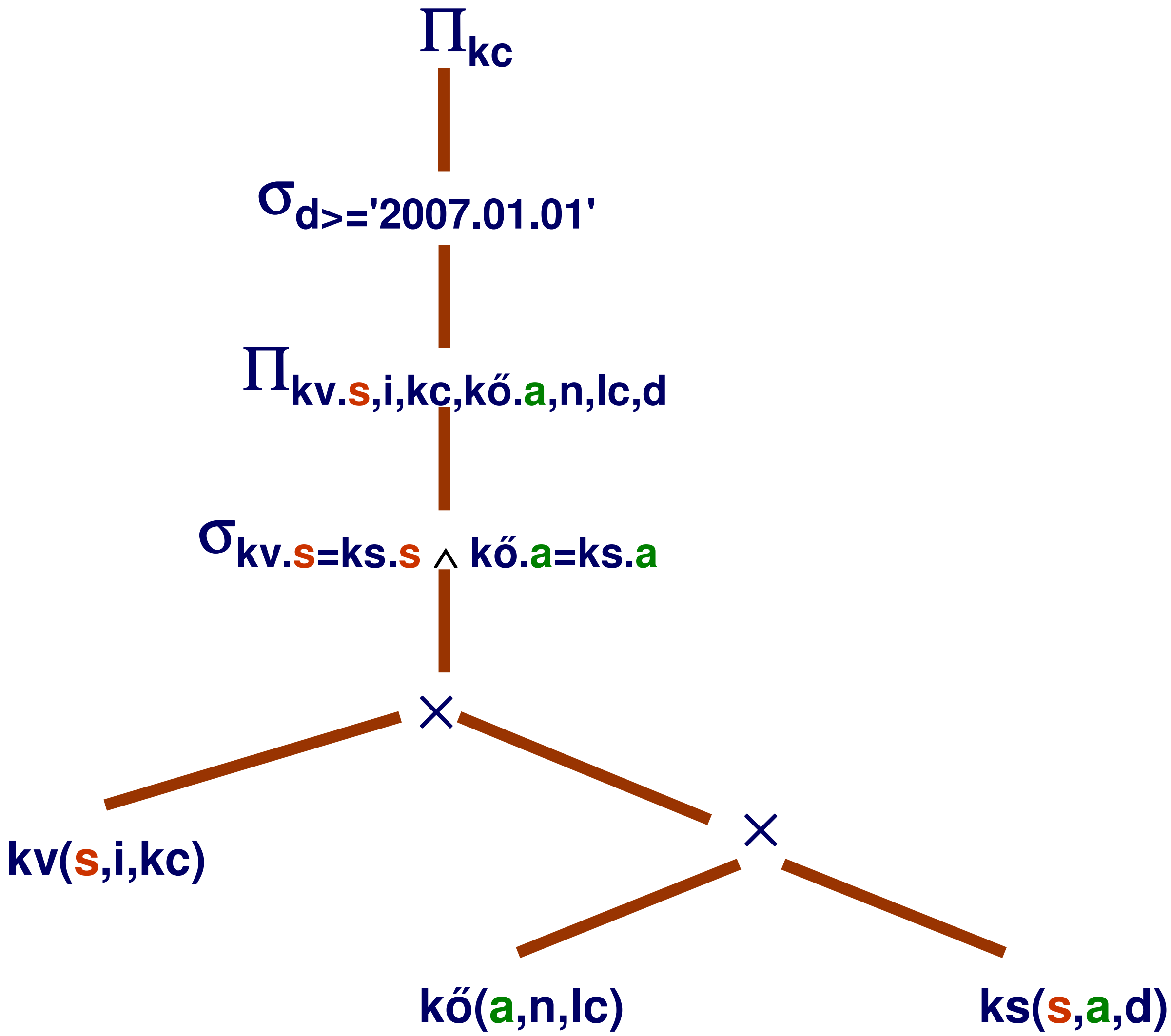
Algebrai optimalizáció

- könyv(sorszám,író,könyvcím)
  - **kv(s,i,kc)**
- kölcsönző(azonosító,név,lakcím)
  - **kő(a,n,lc)**
- kölcsönzés(sorszám,azonosító,dátum)
  - **ks(s,a,d)**
- Milyen című könyveket kölcsönöztek ki ebben az évben?
- $\Pi_{\mathbf{kc}}(\sigma_{\mathbf{d>='2007.01.01'}}(\mathbf{kv}|\times|\mathbf{kő}|\times|\mathbf{ks}))$
- Az összekapcsolásokat valamilyen sorrendben kifejezzük az alapműveletekkel:

$$\Pi_{\mathbf{kc}}(\sigma_{\mathbf{d>='2007.01.01'}}(\Pi_{\mathbf{kv.s,i,kc,kő.a,n,lc,d}}(\sigma_{\mathbf{kv.s=ks.s} \wedge \mathbf{kő.a=ks.a}}(\mathbf{kv}\times(\mathbf{kő}\times\mathbf{ks})))))$$

Algebrai optimalizáció

$$\Pi_{kc}(\sigma_{d>='2007.01.01'}(\Pi_{kv.s,i,kc,kő.a,n,lc,d}(\sigma_{kv.s=ks.s \wedge kő.a=ks.a}(kv\times(kő\times ks)))))$$



Algebrai optimalizáció

- E1(r1,...,rk) és E2(r1,...,rk) **relációs algebrai kifejezések ekvivalensek** (E1 $\cong$ E2), ha tetszőleges r1,...,rk relációkat véve E1(r1,...,rk)=E2(r1,...,rk).
- 11 szabályt adunk meg. A szabályok olyan állítások, amelyek kifejezések ekvivalenciáját fogalmazzák meg. Bizonyításuk könnyen végiggondolható.
- Az állítások egy részében a kifejezések szintaktikus helyessége egyben elégséges feltétele is az ekvivalenciának.

1. **Kommutativitás (szorzás, természetes összekapcsolás, téta-összekapcsolás)**
   - **E1 $\times$ E2 $\cong$ E2 $\times$ E1**
   - **E1 $|\times|$ E2 $\cong$ E2 $|\times|$ E1**
   - **E1 $\underset{\Theta}{|\times|}$ E2 $\cong$ E2 $\underset{\Theta}{|\times|}$ E1**

Algebrai optimalizáció

2. **Asszociativitás (szorzás, természetes összekapcsolás, téta-összekapcsolás)**
   - $(E1 \times E2) \times E3 \cong E2 \times (E1 \times E3)$
   - $(E1 \mid\times\mid E2) \mid\times\mid E3 \cong E1 \mid\times\mid (E2 \mid\times\mid E3)$
   - $(E1 \underset{\Theta}{\mid\times\mid} E2) \underset{\Theta}{\mid\times\mid} E3 \cong E1 \underset{\Theta}{\mid\times\mid} (E2 \underset{\Theta}{\mid\times\mid} E3)$
3. **Vetítések összevonása, bővítése**
   - Legyen $\underline{A}$ és $\underline{B}$ két részhalmaza az E reláció oszlopainak úgy, hogy $\underline{\mathbf{A}} \subseteq \underline{\mathbf{B}}$.
   - Ekkor $\Pi_{\underline{A}}(\Pi_{\underline{B}}(E)) \cong \Pi_{\underline{A}}(E)$.
4. **Kiválasztások felcserélhetősége, felbontása**
   - Legyen F1 és F2 az E reláció oszlopain értelmezett kiválasztási feltétel.
   - Ekkor $\sigma_{F1 \wedge F2}(E) \cong \sigma_{F1}(\sigma_{F2}(E)) \cong \sigma_{F2}(\sigma_{F1}(E))$.

Algebrai optimalizáció

5. **Kiválasztás és vetítés felcserélhetősége**

- Legyen F az E relációnak csak az A oszlopain értelmezett kiválasztási feltétel.

a) • Ekkor $\Pi_{\underline{A}}(\sigma_F(E)) \cong \sigma_F(\Pi_{\underline{A}}(E))$.

- Általánosabban: Legyen F az E relációnak csak az $\underline{A} \cup \underline{B}$ oszlopain értelmezett kiválasztási feltétel, ahol $\underline{A} \cap \underline{B} = \emptyset$.

b) • Ekkor $\Pi_{\underline{A}}(\sigma_F(E)) \cong \Pi_{\underline{A}}(\sigma_F(\Pi_{\underline{A} \cup \underline{B}}(E)))$.

6. **Kiválasztás és szorzás felcserélhetősége**

- Legyen F az E1 reláció oszlopainak egy részhalmazán értelmezett kiválasztási feltétel.

a) • Ekkor $\sigma_F(E1 \times E2) \cong \sigma_F(E1) \times E2$.

- Speciálisan: Legyen i=1,2 esetén Fi az Ei reláció oszlopainak egy részhalmazán értelmezett kiválasztási feltétel, legyen továbbá $F=F1 \wedge F2$.

b) • Ekkor $\sigma_F(E1 \times E2) \cong \sigma_{F1}(E1) \times \sigma_{F2}(E2)$.

- Általánosabban: Legyen F1 az E1 reláció oszlopainak egy részhalmazán értelmezett kiválasztási feltétel, legyen F2 az E1×E2 reláció oszlopainak egy részhalmazán értelmezett kiválasztási feltétel, úgy hogy mindkét sémából legalább egy oszlop szerepel benne, legyen továbbá $F=F1 \wedge F2$.

c) • Ekkor $\sigma_F(E1 \times E2) \cong \sigma_{F2}(\sigma_{F1}(E1) \times E2)$.

Algebrai optimalizáció

7. **Kiválasztás és egyesítés felcserélhetősége**
   - Legyen E1, E2 relációk sémája megegyező, és F a közös sémán értelmezett kiválasztási feltétel.
   - Ekkor $\boldsymbol{\sigma_F(E1 \cup E2) \cong \sigma_F(E1) \cup \sigma_F(E2)}$.
8. **Kiválasztás és kivonás felcserélhetősége**
   - Legyen E1, E2 relációk sémája megegyező, és F a közös sémán értelmezett kiválasztási feltétel.
   - Ekkor $\boldsymbol{\sigma_F(E1 - E2) \cong \sigma_F(E1) - \sigma_F(E2)}$.
9. **Kiválasztás és természetes összekapcsolás felcserélhetősége**
   - Legyen F az E1 és E2 közös oszlopainak egy részhalmazán értelmezett kiválasztási feltétel.
   - Ekkor $\boldsymbol{\sigma_F(E1 |\times| E2) \cong \sigma_F(E1) \; |\times| \; \sigma_F(E2)}$.

Algebrai optimalizáció

10. Vetítés és szorzás felcserélhetősége

- Legyen i=1,2 esetén $\underline{Ai}$ az Ei reláció oszlopainak egy halmaza, valamint legyen $\underline{A}=\underline{A1}\cup\underline{A2}$.
- Ekkor $\Pi_{\underline{A}}(\mathbf{E1}\times\mathbf{E2}) \cong \Pi_{\underline{A1}}(\mathbf{E1})\times\Pi_{\underline{A2}}(\mathbf{E2})$.

11. Vetítés és egyesítés felcserélhetősége

- Legyen E1 és E2 relációk sémája megegyező, és legyen $\underline{A}$ a sémában szereplő oszlopok egy részhalmaza.
- Ekkor $\Pi_{\underline{A}}(\mathbf{E1}\cup\mathbf{E2}) \cong \Pi_{\underline{A}}(\mathbf{E1})\cup\Pi_{\underline{A}}(\mathbf{E2})$.

- Megjegyzés: A vetítés és kivonás nem cserélhető fel, azaz $\Pi_{\underline{A}}(\mathbf{E1} - \mathbf{E2}) \not\cong \Pi_{\underline{A}}(\mathbf{E1}) - \Pi_{\underline{A}}(\mathbf{E2})$. Például:

E1:

A	B
0	0
0	1

E2:

A	B
0	0

esetén $\Pi_{\mathbf{A}}(\mathbf{E1} - \mathbf{E2})$:

A
0

míg

$\Pi_{\mathbf{A}}(\mathbf{E1}) - \Pi_{\mathbf{A}}(\mathbf{E2}) = \emptyset$

Algebrai optimalizáció

- Az optimalizáló algoritmus a következő **heurisztikus elveken** alapul:

1. **Minél hamarabb szelektáljunk**, hogy a részkifejezések várhatóan kisebb relációk legyenek.
2. A szorzás utáni kiválasztásokból **próbáljunk természetes összekapcsolásokat képezni**, mert az összekapcsolás hatékonyabban kiszámolható, mint a szorzatból történő kiválasztás.
3. **Vonjuk össze az egymás utáni unáris műveleteket** (kiválasztásokat és vetítéseket), és ezekből lehetőleg egy kiválasztást, vagy vetítést, vagy kiválasztás utáni vetítést képezzünk. Így csökken a műveletek száma, és általában a kiválasztás kisebb relációt eredményez, mint a vetítés.
4. **Keressünk közös részkifejezéseket**, amiket így elég csak egyszer kiszámolni a kifejezés kiértékelése során.

Algebrai optimalizáció

- **Algebrai optimalizációs algoritmus:**
- INPUT: relációs algebrai kifejezés kifejezésfája
- OUTPUT: optimalizált kifejezésfa optimalizált kiértékelése

Hajtsuk végre az alábbi lépéseket a megadott sorrendben:

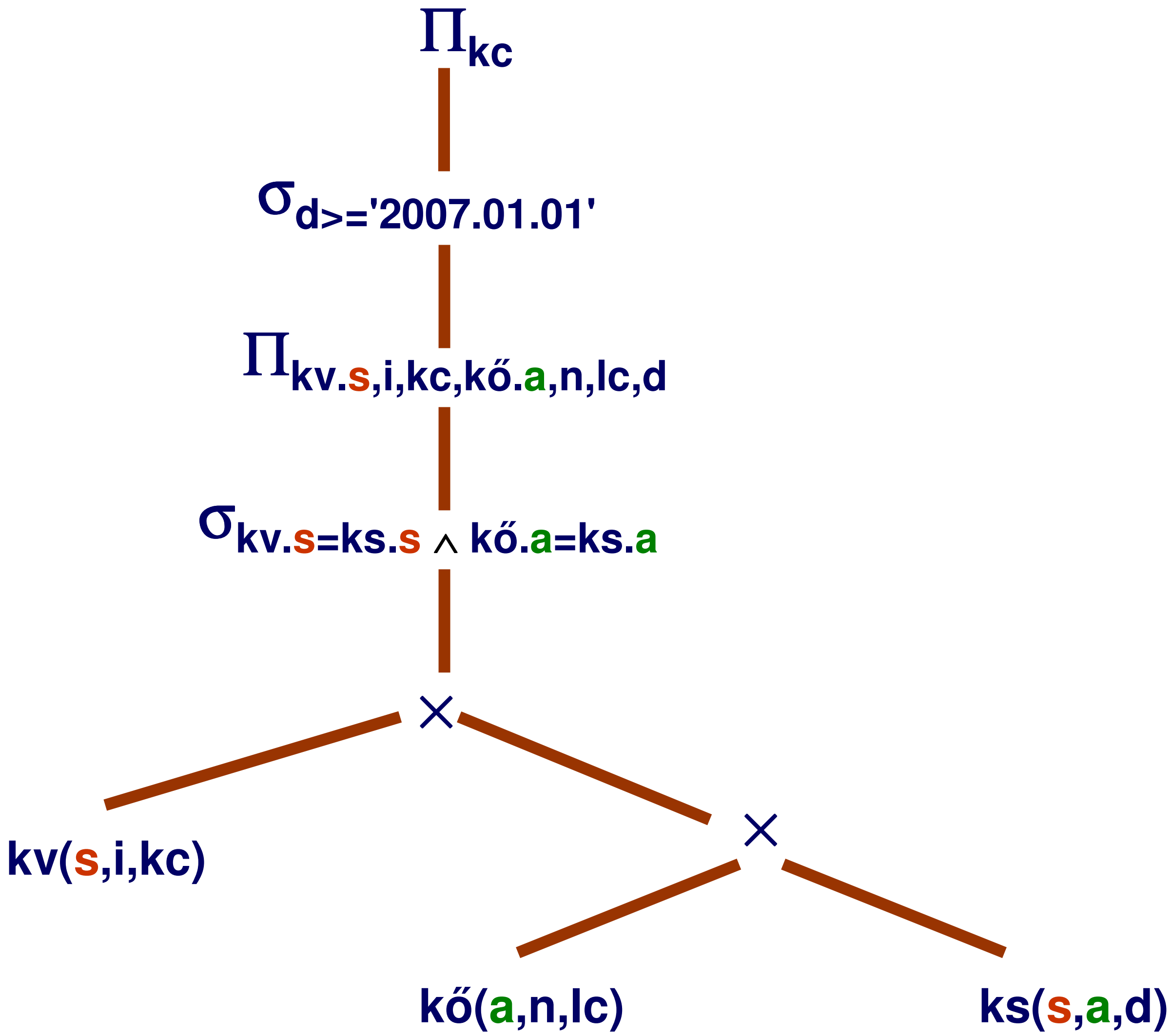
1. **A kiválasztásokat bontsuk fel** a **4. szabály** segítségével:
   - $\sigma_{F1\wedge\ldots\wedge Fn}(E) \cong \sigma_{F1}(\ldots(\sigma_{Fn}(E)))$
2. **A kiválasztásokat** a **4., 5., 6., 7., 8., 9. szabályok** segítségével **vigyük** olyan **mélyre** a kifejezésfában, amilyen mélyre csak lehet.
3. **A vetítéseket** a **3., 5., 10., 11. szabályok** segítségével **vigyük** olyan **mélyre** a kifejezésfában, amilyen mélyre csak lehet. Hagyjuk el a triviális vetítéseket, azaz az olyanokat, amelyek az argumentum reláció összes attribútumára vetítenek.
4. Ha egy relációs változóra vagy konstans relációra közvetlenül egymás után kiválasztásokat vagy vetítéseket alkalmazunk, akkor ezeket a **3., 4., 5. szabályok** segítségével **vonjuk össze egy kiválasztássá, vagy egy vetítéssé, vagy egy kiválasztás utáni vetítéssé, ha lehet** (azaz egy $\Pi.(\sigma.())$ alakú kifejezéssé). **Ezzel megkaptuk az optimalizált kifejezésfát.**
5. A gráfot **a bináris műveletek alapján bontsuk részgráfokra**. Minden részgráf egy bináris műveletnek feleljen meg. A részgráf csúcsai legyenek: a bináris műveletnek $(\cup, —, \times)$ megfelelő csúcs és a csúcs felett a következő bináris műveletig szereplő kiválasztások (σ) és vetítések (Π). Ha a bináris művelet szorzás ($\times$), és a részgráf equi-joinnak felel meg, és a szorzás valamelyik ága nem tartalmaz bináris műveletet, akkor ezt az ágat is vegyük hozzá a részgráfhoz.
6. Az előző lépésben kapott részgráfok is fát képeznek. **Az optimális kiértékeléshez** ezt a fát értékeljük ki alulról felfelé haladva, tetszőleges sorrendben.

Megjegyzés. Az **equi-join** azt jelenti, hogy a kiválasztás feltétele egyenlőség, amely a szorzás két ágának egy-egy oszlopát hasonlítja össze.

Algebrai optimalizáció

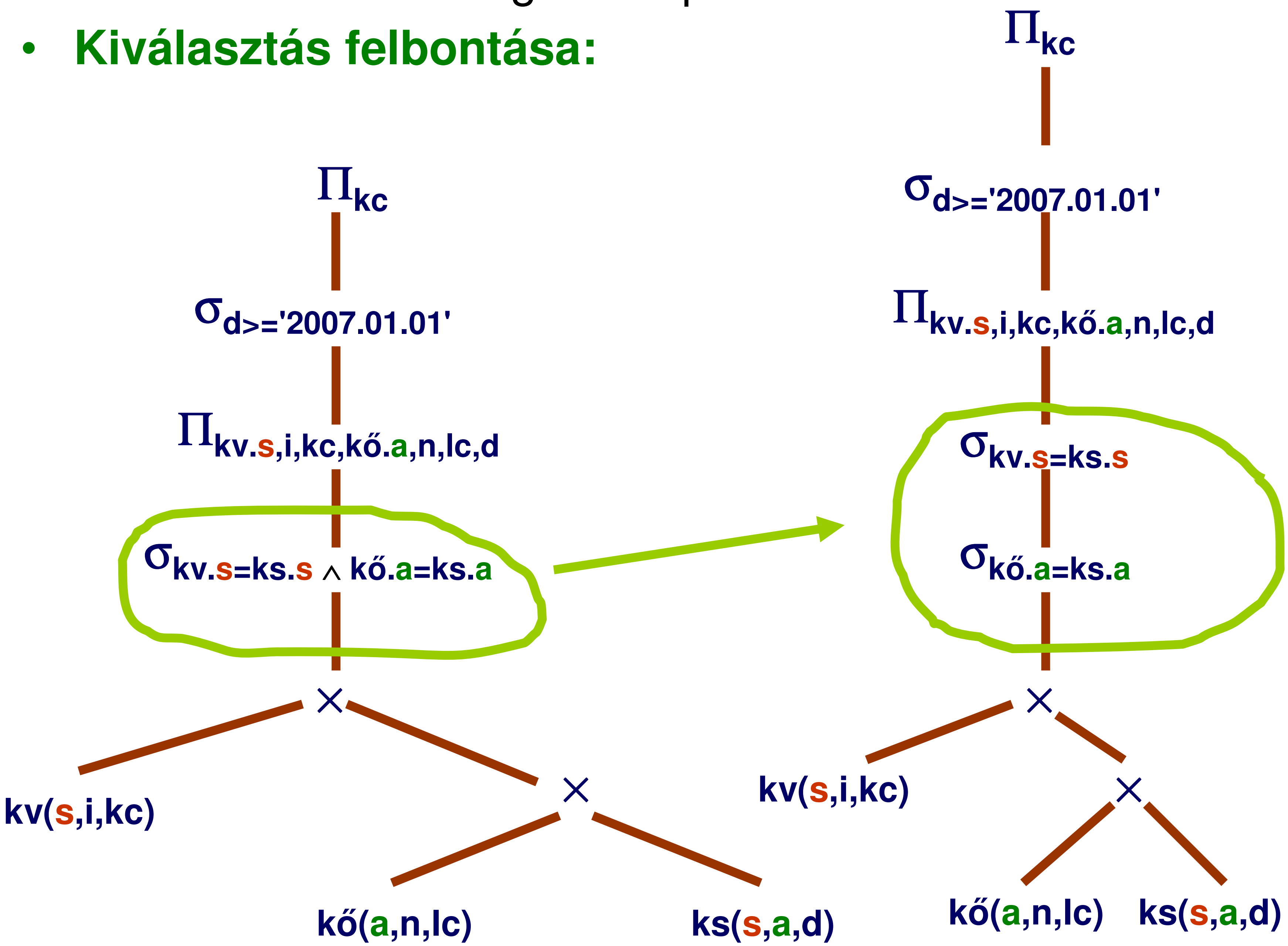
- Optimalizáljuk a következő kifejezést:

$$\Pi_{kc}(\sigma_{d>='2007.01.01'}(\Pi_{kv.s,i,kc,kő.a,n,lc,d}(\sigma_{kv.s=ks.s \wedge kő.a=ks.a}(kv \times (kő \times ks)))))$$

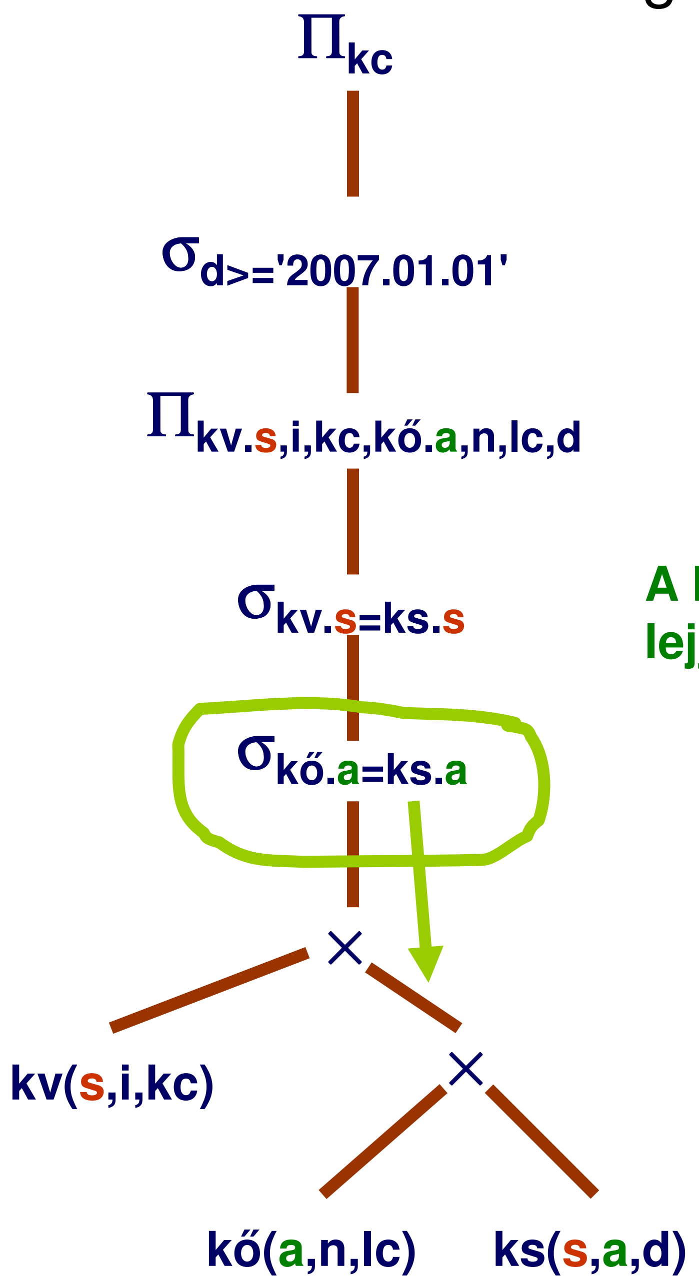


Algebrai optimalizáció

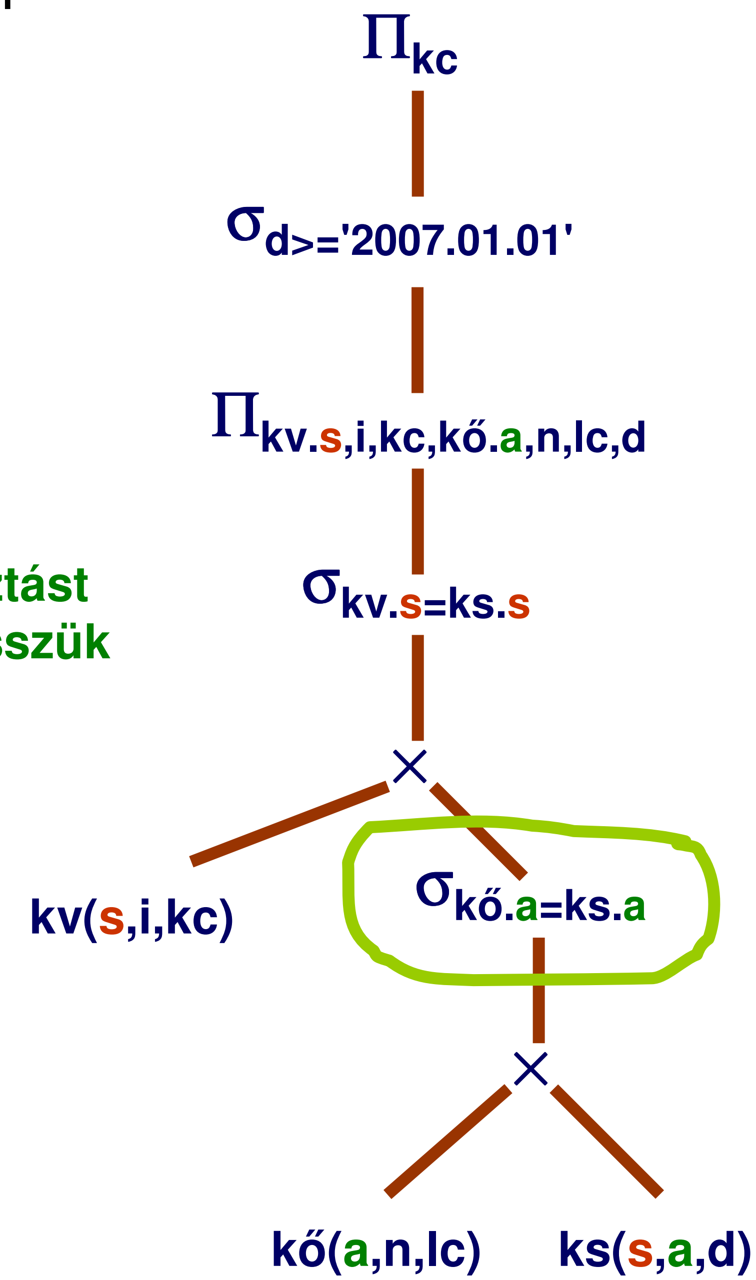
- **Kiválasztás felbontása:**



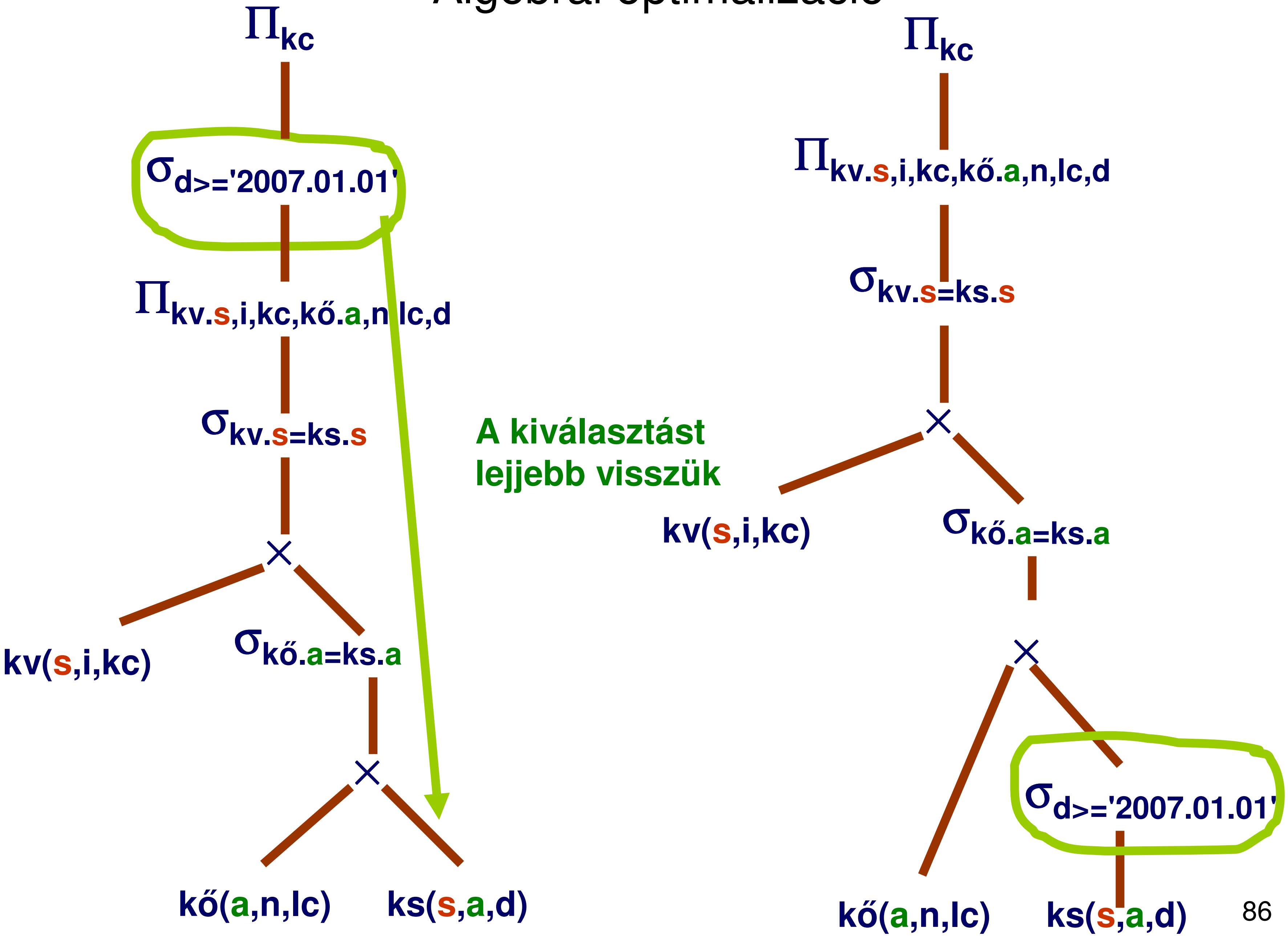
Algebrai optimalizáció



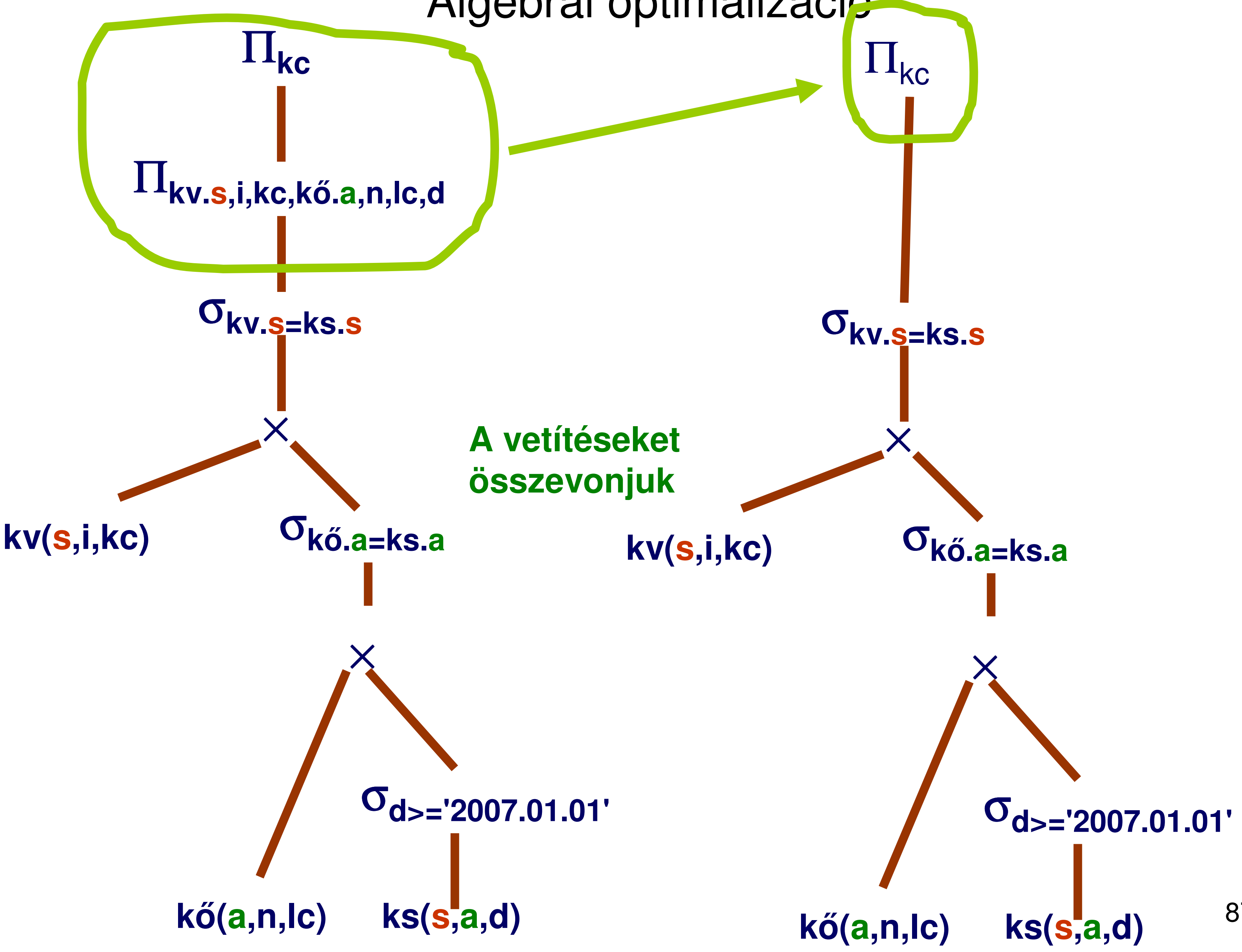
A kiválasztást lejjebb visszük



Algebrai optimalizáció

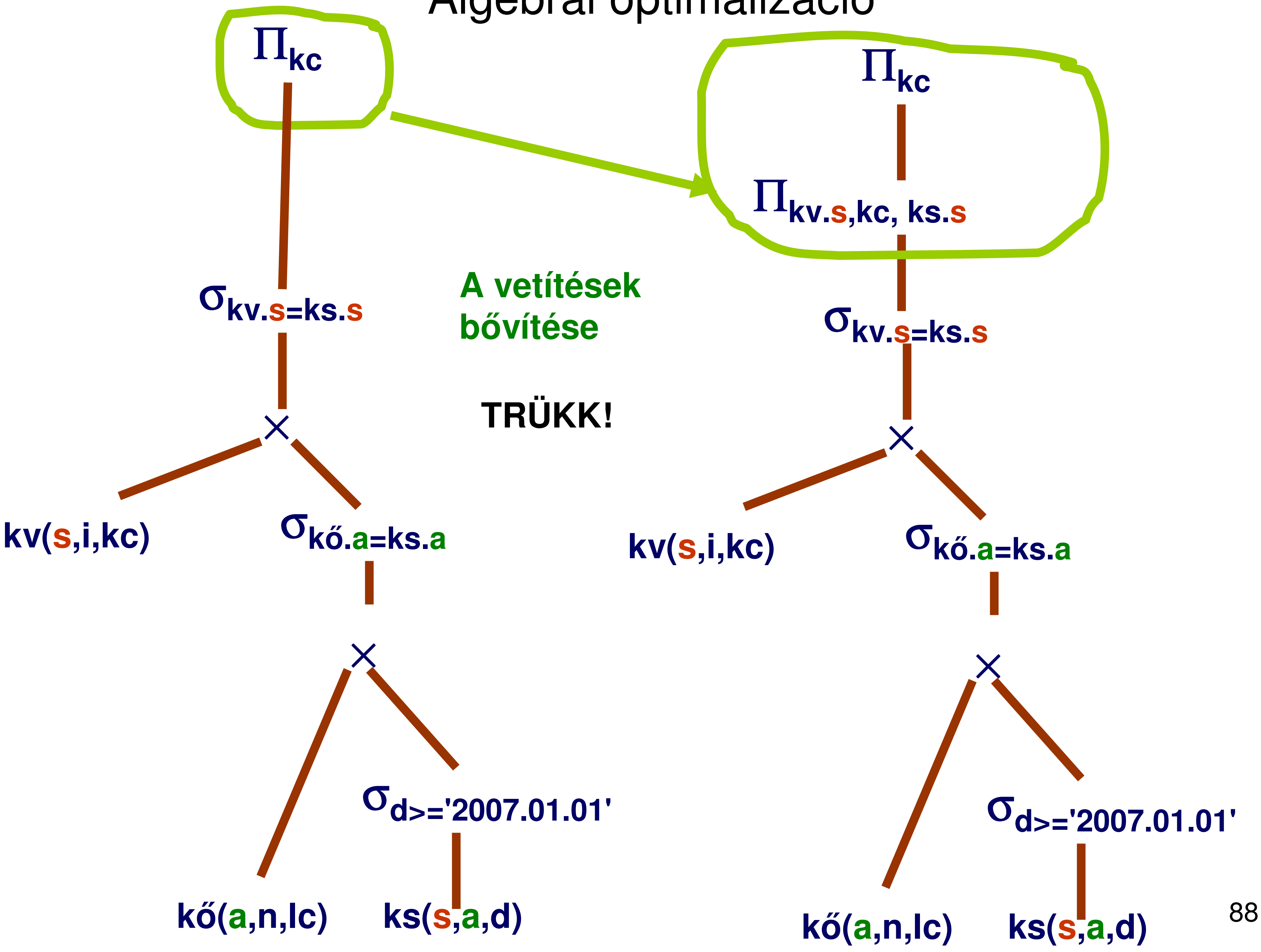


Algebrai optimalizáció

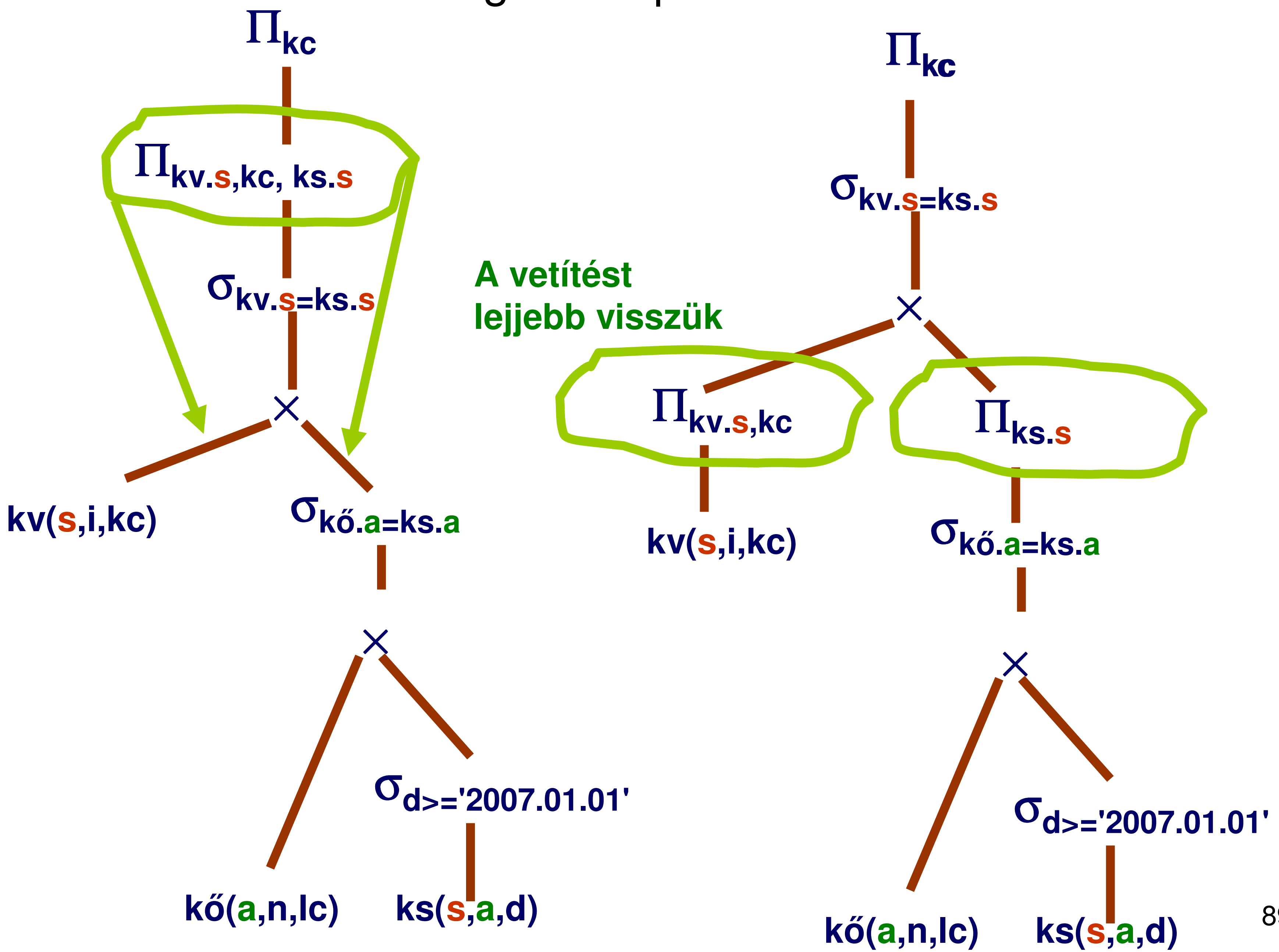


A vetítéseket összevonjuk

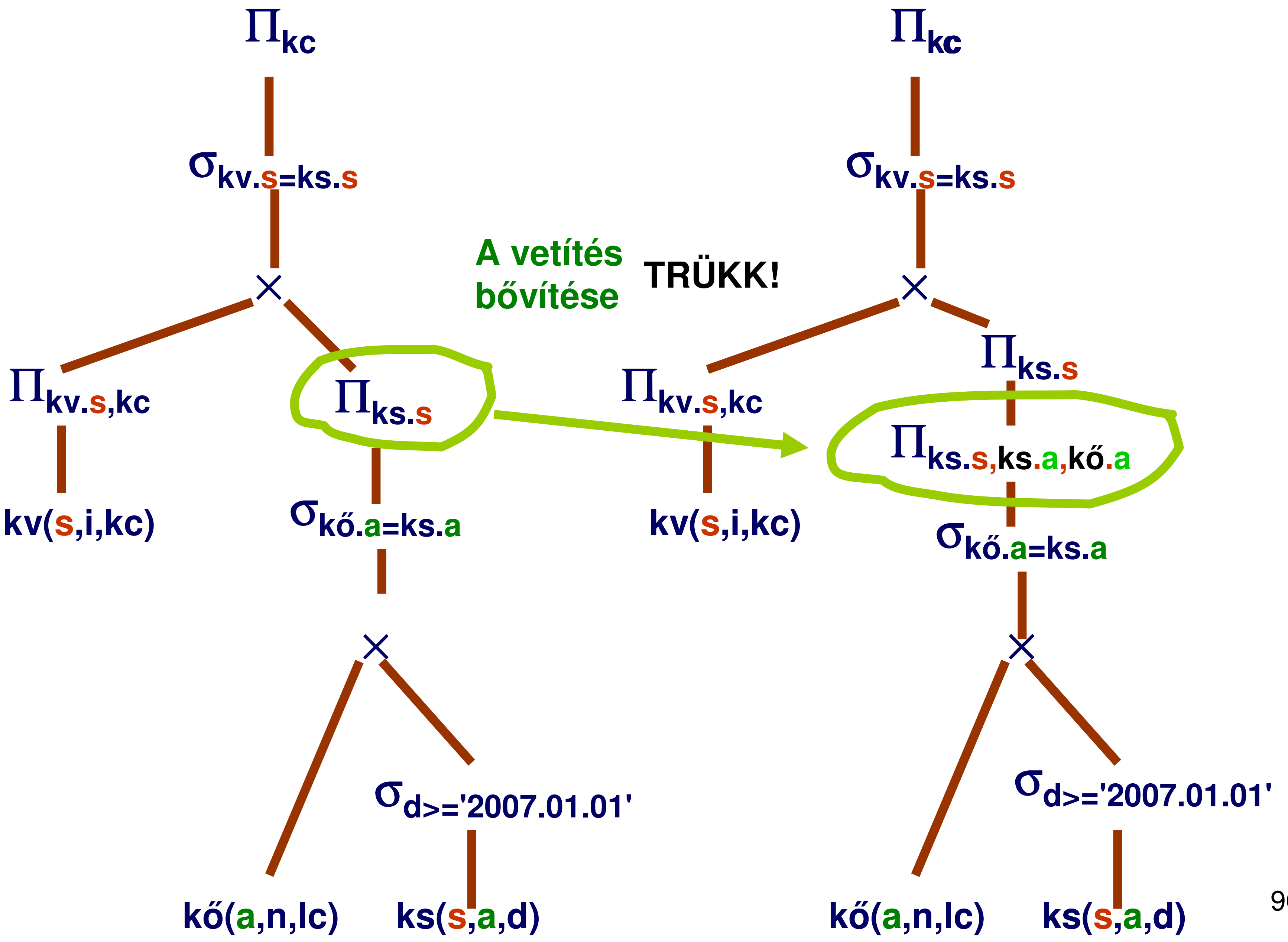
Algebrai optimalizáció



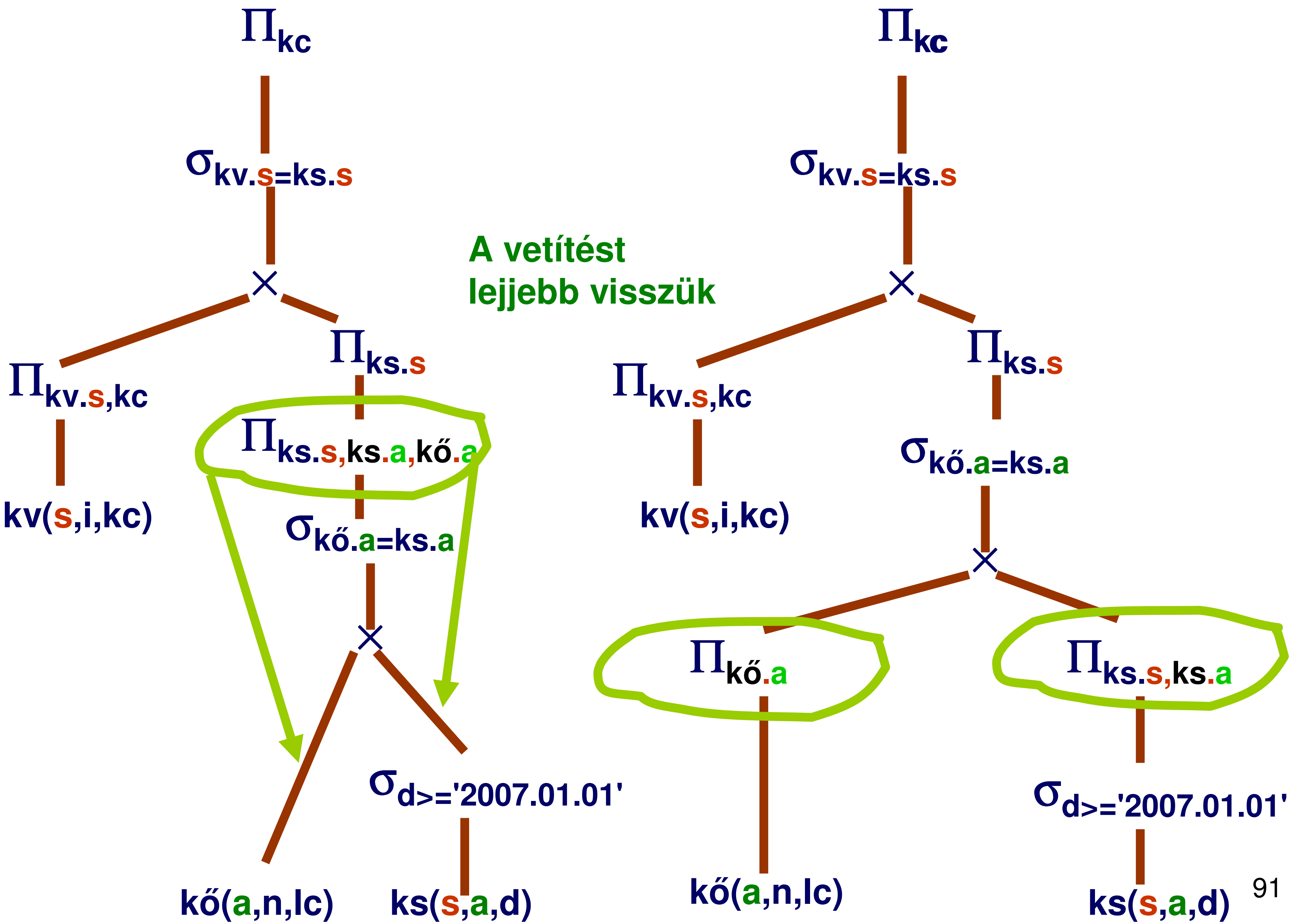
Algebrai optimalizáció



Algebrai optimalizáció

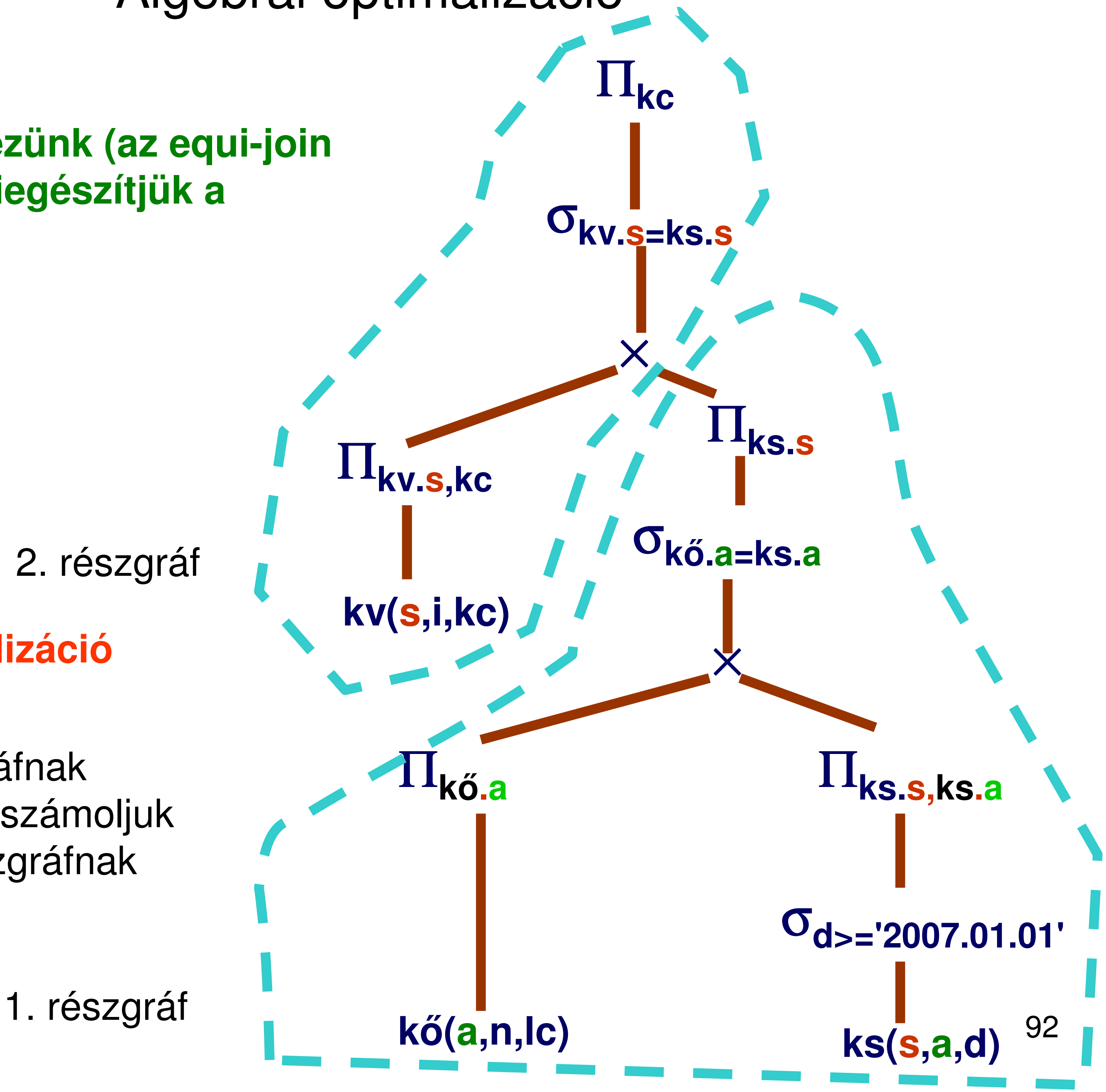


Algebrai optimalizáció



Algebrai optimalizáció

Részgráfokat képezünk (az equi-join miatt a levelekig kiegészítjük a csoportokat)



Az algebrai optimalizáció eredménye:

Először az 1. részgráfnak megfelelő kifejezést számoljuk ki, és utána a 2. részgráfnak megfelelő kifejezést.