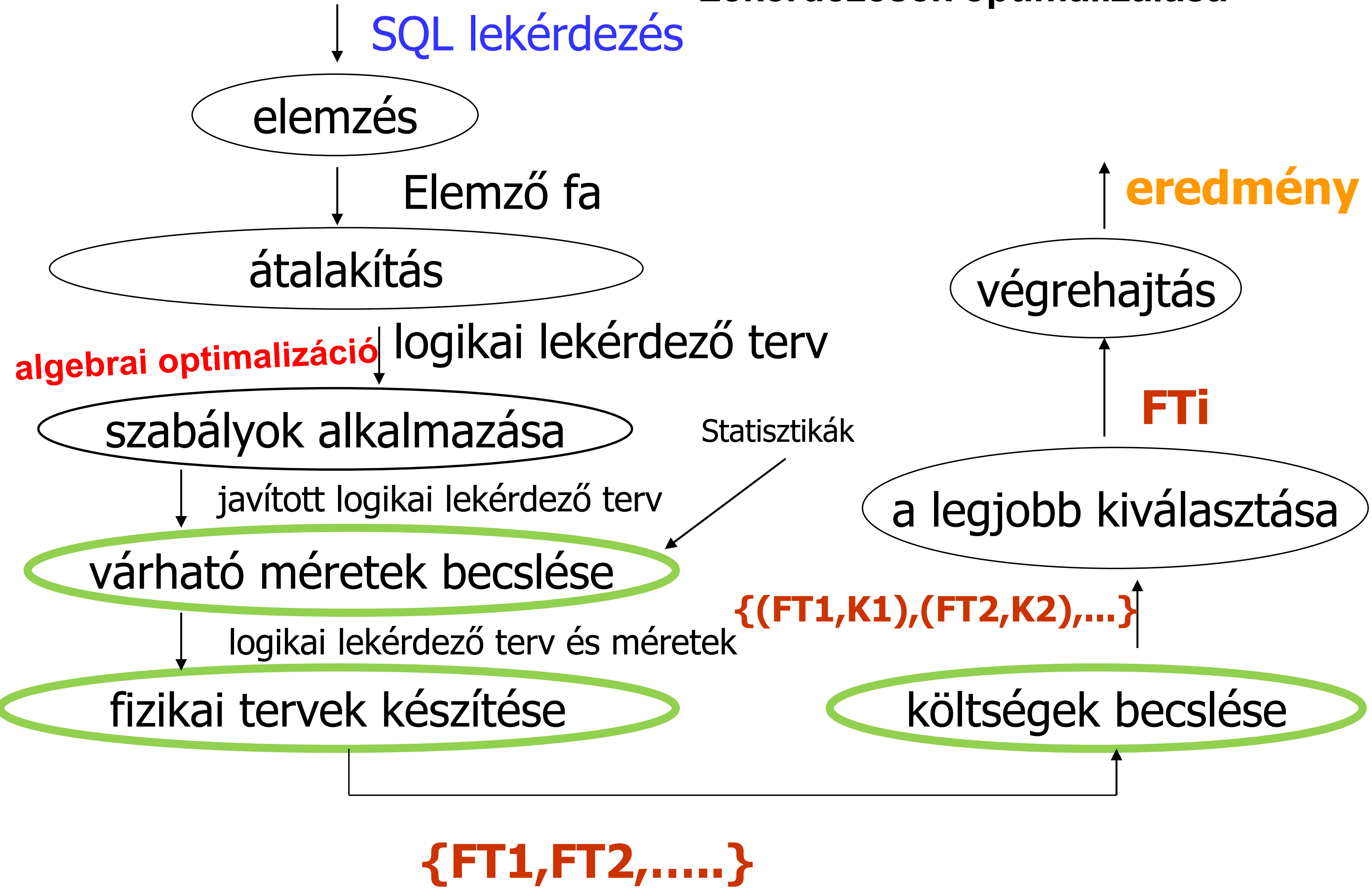
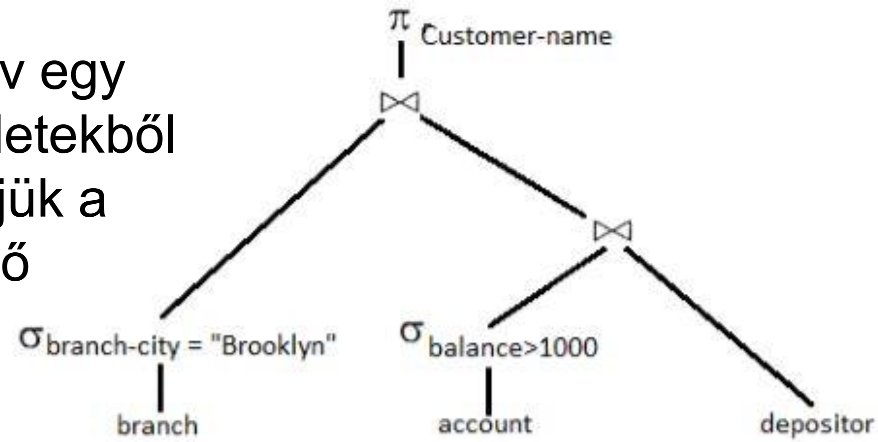
Fizikai tervek (folytatás)

- Paraméterek, költségek
- Fizikai fájlszervezés,
  - szekvenciális (kupac), hasító indexek (statikus, dinamikus (kiterjeszthető, lineáris)
  - Rendezett állomány, elsődleges, másodlagos indexek, többszintű indexek, B^+-fák, B*-fák
- Műveletek megvalósítása, kiszámítási költség, outputméret
- Optimális fizikai terv meghatározása

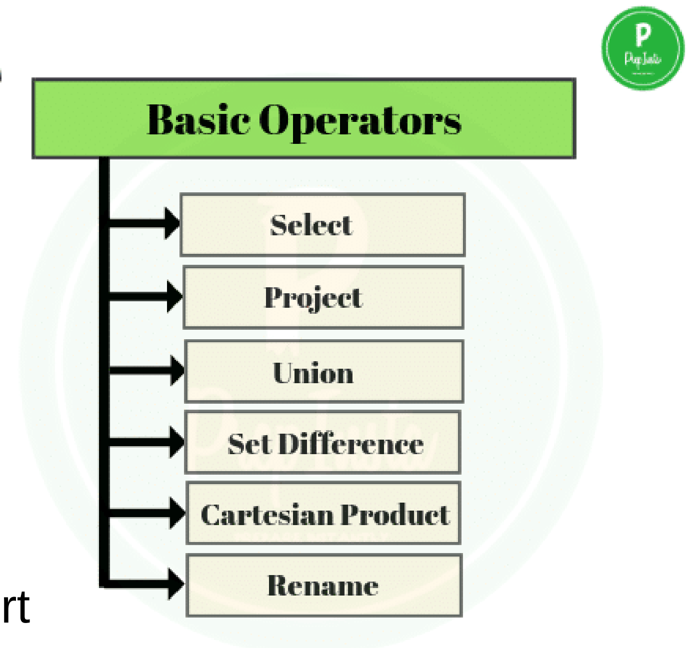
Lekérdezések optimalizálása



- Az optimális logikai lekérdező terv egy relációs algebrai kifejezés. Műveletekből áll. Ha minden műveletnek ismerjük a költségét, akkor a fának megfelelő költséget is ki tudjuk számolni.



- Hogyan végezzük el az egyes műveletek? Adott inputméret esetén mekkora a várható output méret.
- Műveletek:
  - kiválasztás: **σ** (szekvenciális, index, rendezés)
  - vetítés: **π**
  - unió: ∪
  - különbség: **-**
  - szorzat: **x**
  - átnevezés: ρ (ezzel nem foglalkozunk, mert méretet nem változtat, költsége nincs)
  - összekapcsolás: ⋈ (származtatott, de fontos művelet)



- SELECT * FROM student WHERE name=Paul
  - $\sigma_{name=Paul}(student)$
- $\pi_{name}(\sigma_{cid<00112235}(student))$
- $\pi_{name}(\sigma_{coursename=Advanced\ DBs}((student \bowtie_{cid} takes) \bowtie_{courseid} course))$
- Sokféle lehetőségünk van egy lekérdezés kiértékelésére

  $\pi_{name}(\sigma_{coursename=Advanced\ DBs}((student \bowtie_{cid} takes) \bowtie_{courseid} course))$
  - $\pi_{name}((student \bowtie_{cid} takes) \bowtie_{courseid} \sigma_{coursename=Advanced\ DBs}(course)))$
  - (relációs algebrai optimalizálás – várhatóan jobb költség - heurisztika)

  Most viszont számokkal kifejezett költséget tudunk összehasonlítani.

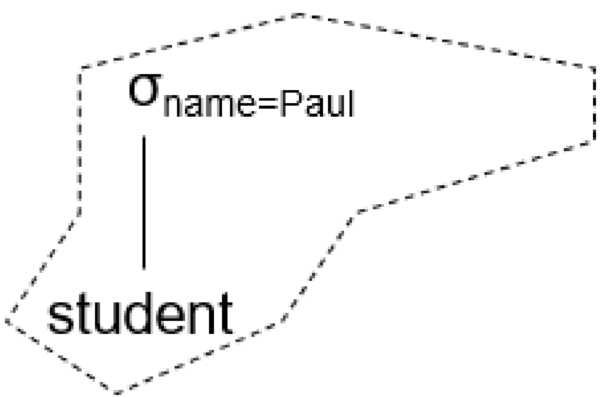
student	
cid	name
00112233	Paul
00112238	Rob
00112235	Matt

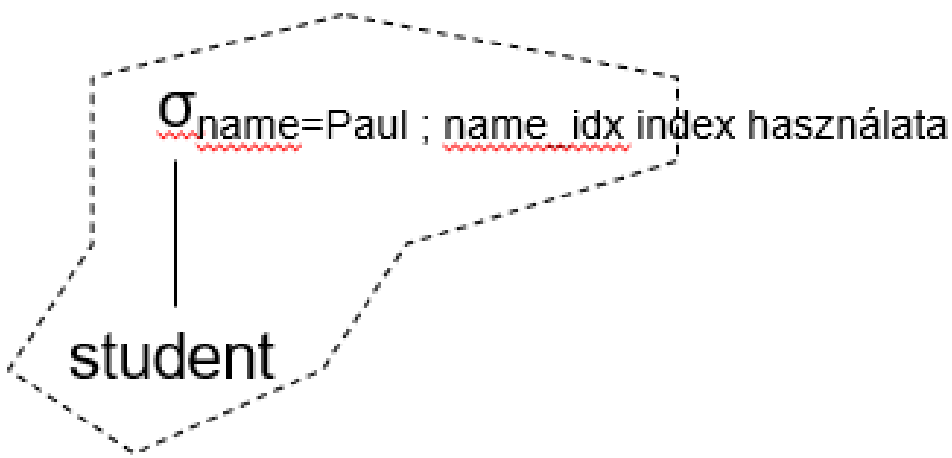
takes	
cid	courseid
00112233	312
00112233	395
00112235	312

course	
courseid	coursename
312	Advanced DBs
395	Machine Learning

Műveletek kiértékelésének meghatározása

- Több lehetőség egy művelet elvégzésére
  - $\sigma_{name=Paul}(student)$
    - fájlban szekvenciális keresés
    - másodlagos index a student.name mezőn
- Több elérési útvonal
  - elérési útvonal: mely módon érhetjük el a rekordokat (például a name_idx index alapján)



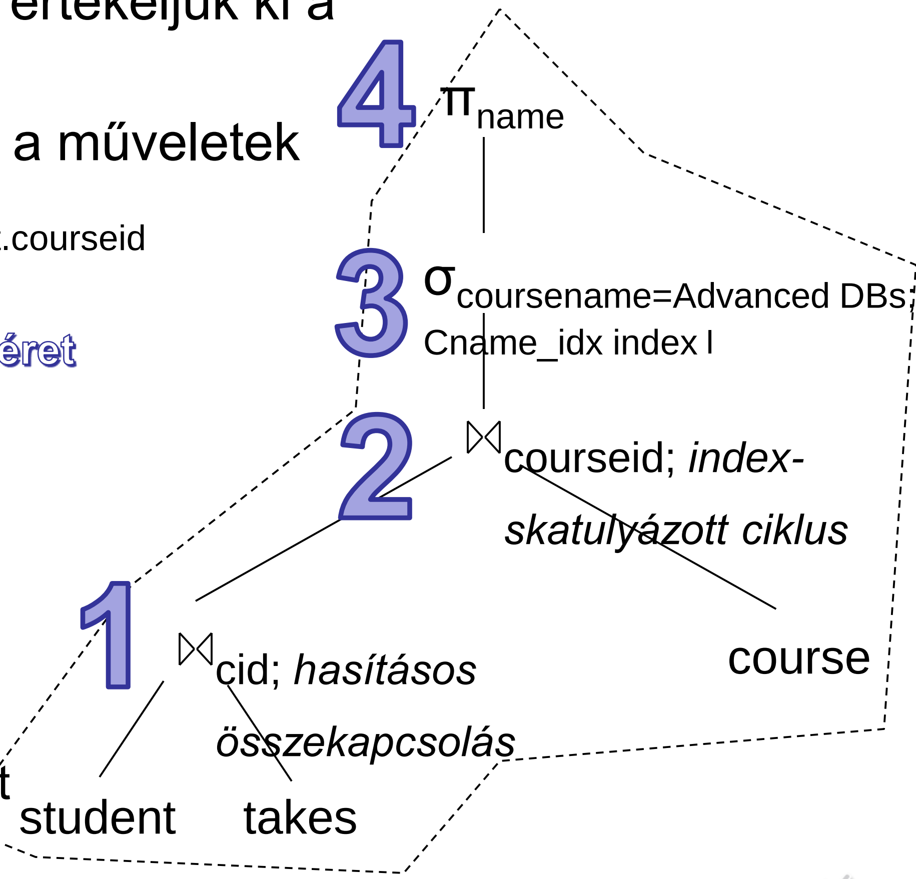


$\pi_{name}(\sigma_{coursename=Advanced\ DBs}((student \bowtie_{cid} takes) \bowtie_{courseid} course))$

- Adjuk meg, melyik elérési útvonalat használjuk
- Adjuk meg, milyen algoritmussal értékeljük ki a műveleteket
- Adjuk meg, hogyan következnek a műveletek
- **INPUT**: T_s, T_t, T_c, bf_s, bf_t, bf_c, $I_{s.cid}$, $I_{t.courseid}$
- K_1, O_1
- K_2, O_2
- K_3, O_3
- K_4, O_4

Számítási költség, outputméret

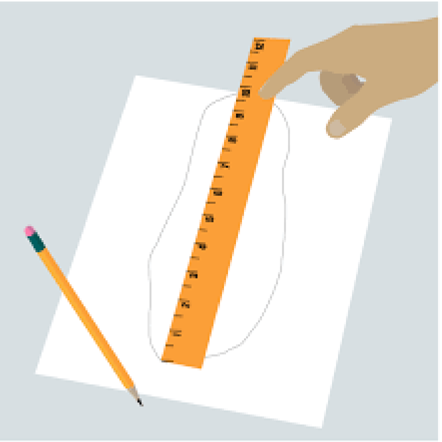
- **OUTPUT**: $\mathbf{K} = K_1 + K_2 + K_3 + K_4 + O_4$
- Optimalizáció:
  - becsüljük meg a tervek költségét (nem mindet)
  - válasszuk a legalacsonyabb becsült költségűt



Költségbecslés

- Mit kell számításba venni:
  - **Lemez I/O**
    - szekvenciális
    - Indexelt elérés
  - CPU idő (elhanyagolható)
  - Hálózati kommunikáció (csak osztott adatbázisok esetén)
- Mit fogunk figyelembe venni:
  - Lemez I/O
    - Lapok (blokkok) olvasása, írása
  - Elhanyagoljuk a végeredmény kiírásának költségét (mivel amiket összehasonlítunk, azoknál mindnél ugyanaz a végeredmény mérete) az előző példában $\mathbf{K}=K_1+K_2+K_3+K_4+O_4$ helyett $\mathbf{K}=K_1+K_2+K_3+K_4$ –t elég vizsgálnunk a fizikai tervek összehasonlításánál.

Műveletek és költségek



SQL | 0.052 seconds

OPERATION	OBJECT_NAME	CARDINALITY	COST
SELECT STATEMENT		106	6
MERGE JOIN		106	6
TABLE ACCESS (BY INDEX ROWID)	DEPARTMENTS	27	2
INDEX (FULL SCAN)	DEPT_ID_PK	27	1
SORT (JOIN)		107	4
Access Predicates			
E.DEPARTMENT_ID=D.DEPARTMENT_ID			
Filter Predicates			
E.DEPARTMENT_ID=D.DEPARTMENT_ID			
TABLE ACCESS (FULL)	EMPLOYEES	107	3

Műveletek és költségek

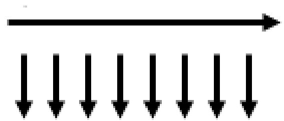
- Műveletek: **σ, π, ∪, -, x**, **⋈, ∩,**
- Költségek:
  - **N_R**: R rekordjainak száma (ezt korábban T_R-rel jelöltük)
  - **L_R**: R egy rekordjának mérete (ezt korábban l_R-rel jelöltük)
  - **F_R**: blokkolási tényező (ezt korábban bf_R-rel jelöltük)
    - egy lapon, blokkban levő rekordok száma
  - **B_R**: az R reláció tárolásához szükséges lapok, blokkok száma
  - **V(A,R)**: az *A* mező különböző értékeinek száma R-ben

    (Képméret) (ezt korábban $I_{R.A}$-val jelöltük)
  - **SC(A,R)**: az *A* mező kiválasztási számossága R-ben

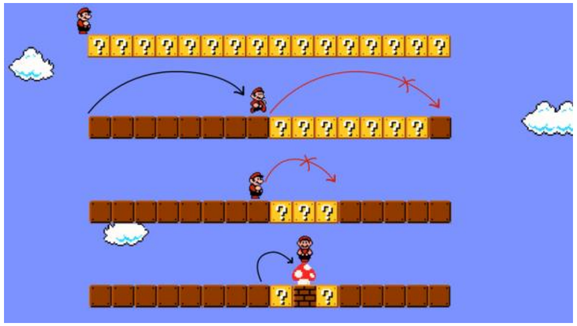
    Szelektivitás – hány darab A=a értékű rekord van (egyenletességi feltétel esetén)
    - *A* kulcs: SC(A,R)=1
    - *A* nem kulcs: SC(A,R)= N_R / V(A,R)
  - **HT_i**: az *i* index szintjeinek száma
  - a törteket és logaritmusokat felfelé kerekítjük

Kiválasztás σ

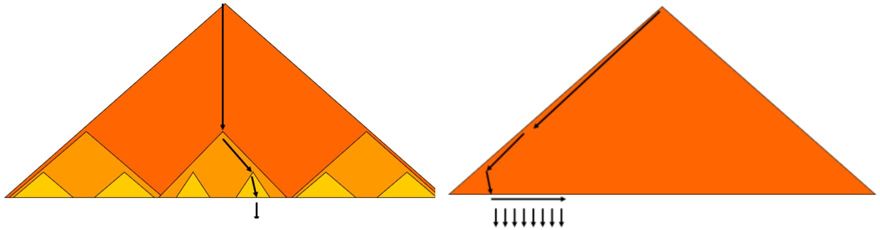
- Lineáris keresés
  - olvassunk be minden lapot és keressük az egyezéseket (egyenlőség vizsgálata esetén)
  - átlagos költség:
    - nem kulcs esetén B_R, kulcs esetén $0.5*B_R$



- Logaritmikus keresés
  - rendezett mező esetén $|\log_2 B_R| + m$
  - átlagos költség:
    - *m* további lapot kell beolvasni
    - $m = \lceil SC(A,R)/F_R \rceil - 1$

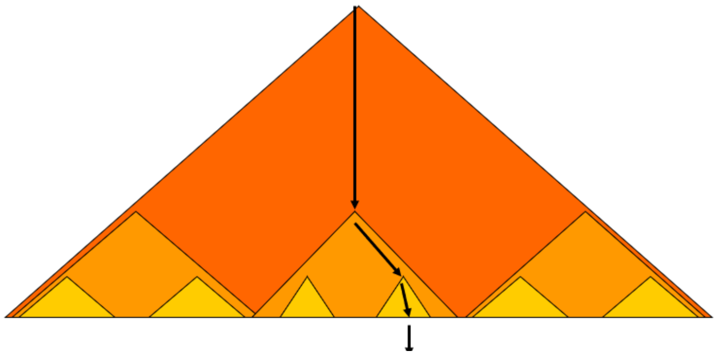


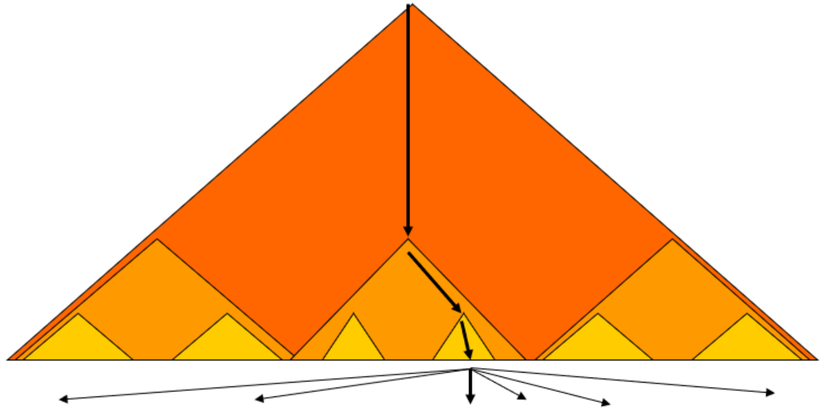
- Elsődleges/cluster index
  - átlagos költség:
    - egyetlen rekord $HT_i + 1$
    - több rekord $HT_i + \lceil SC(A,R)/F_R \rceil$



Kiválasztás σ

- Másodlagos index
  - átlagos költség:
    - kulcs mező $HT_i + 1$
    - nem kulcs mező
      - legrosszabb eset $HT_i + SC(A,R)$
      - a lineáris keresés kedvezőbb, ha sok a megfelelő rekord





Összetett kiválasztás σ_{kif}

- konjunkciós kiválasztás: $\sigma_{\theta_1 \wedge \theta_2 .. \wedge \theta_n}$
  - végezzünk egyszerű kiválasztást a legkisebb költségű θ_i-re
    - pl. a θ_i-hez tartozó index felhasználásával
    - a fennmaradó θ feltételek szerint szűrjük az eredményt
    - $\sigma_{cid > 00112233 \wedge courseid = 312}(takes)$
    - költség: az egyszerű kiválasztás költsége a kiválasztott θ-ra
  - több index
    - válasszuk ki a θ_i-khez tartozó indexeket
    - keressünk az indexekben és adjuk vissza a RID-ket
    - válasz: RID-k metszete
    - költség: a költségek összege + rekordok beolvasása
- diszjunkciós kiválasztás: $\sigma_{\theta_1 \vee \theta_2 .. \vee \theta_n}$
  - több index
    - RID-k uniója
      - költség: a költségek összege + rekordok beolvasása
  - lineáris keresés

Méretbecslés - kiválasztás

- $\sigma_{A=v}(R)$
  - Sorok száma: SC(A,R) Blokkok száma: SC(A,R)/F_R
- $\sigma_{A\leq v}(R)$
  - Sorok száma: $N_R * \frac{v - \min(A,R)}{\max(A,R) - \min(A,R)}$ Blokkok száma: Sorok száma/ F_R
- $\sigma_{\theta_1 \wedge \theta_2 \wedge .. \wedge \theta_n}(R)$
  - szorzódó valószínűségek (s_i az i-ik feltétel szelektivitása:hány rekord elégíti ki)
  - Sorok száma: $N_R * [(s_1/N_R)*(s_2/N_R)*...(s_n/N_R)]$ Blokkok száma: Sorok száma/ F_R
- $\sigma_{\theta_1 \vee \theta_2 v .. \vee \theta_n}(R)$
  - annak valószínűsége, hogy egy rekordra egy θ se igaz:

    $[(1-s_1/N_R)*(1-s_2/N_R)*...*(1-s_n/N_R)]$ s_i az i-ik feltétel szelektivitása
  - Sorok száma: $N_R * (1-[(1-s_1/N_R)*(1-s_2/N_R)*...*(1-s_n/N_R)])$

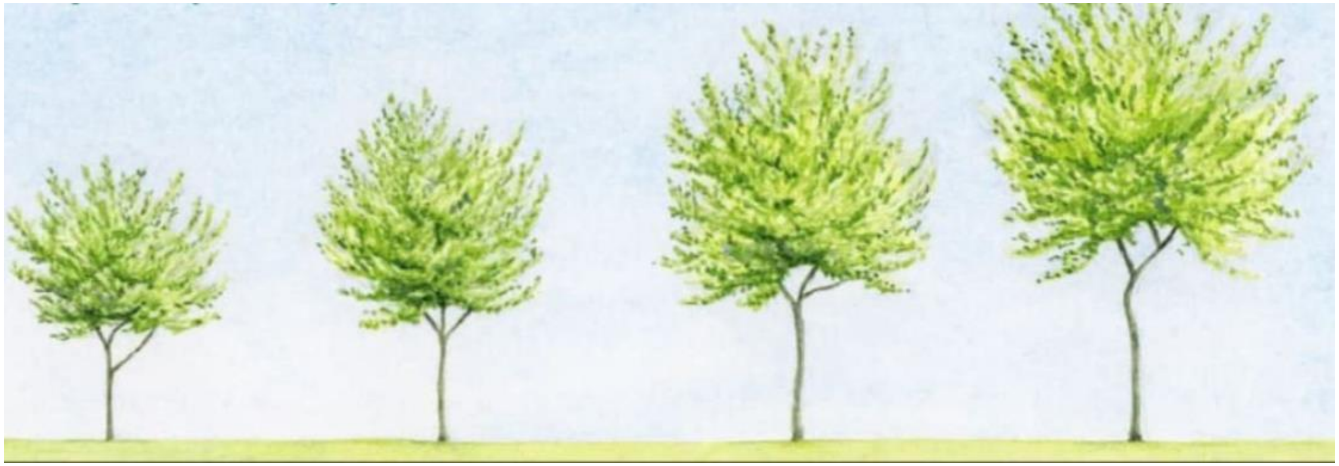
    Blokkok száma: Sorok száma/ F_R

A keresési feltételekre, azaz a bennük szereplő oszlopokra függetlenségi feltételt tettünk fel. Ezzel felső korlátokat kaptunk. Ha nem teljesülne a függetlenség, akkor több egybeesés lehetne, kisebb lenne a méret.

Ezenkívül egyenletességi feltételt is tettünk.

Vetítés és halmazműveletek visszavezetése rendezésre

- SELECT DISTINCT cid FROM takes
  - π-hez szükséges a duplikált értékek kiszűrése
  - Rendezéssel tudjuk a duplikátumokat kiszűrni
- Halmazműveletek
  - R ∩ S ki kell szűrni a duplikált értékeket
  - R ∪ S ki kell szűrni a duplikált értékeket
  - R - S (ha mindkét tábla rendezett, akkor elég egyszerre végigolvasni mindkét táblát.)
  - Mindegyikhez kell a rendezés



Rendezés

- sok művelet hatékony kiértékelése
- a lekérdezés is igényelheti:
  - SELECT cid,name FROM student ORDER BY name
- megvalósítás
  - belső rendezés (ha a rekordok befférnek a memóriába)

  betöltjük: B_R művelet,

  memóriában rendezzük: 0 művelet

  kiírjuk: B_R művelet

  összesen $2*B_R$
  - külső rendezés (ha nem fér be a teljes tábla memóriába): SORT-MERGE

Külső összefésüléses rendezés (1/3)

- Rendező lépés: rendezett futamok létrehozása

  M // a memória mérete blokkokban

  i=0;
  ismétlés

  M lap beolvasása az R relációból a memóriába

  az *M* lap rendezése

  kiírás az R_i fájlba (futamba)

  i növelése

  amíg el nem fogynak a lapok

  N = i // futamok száma

Külső összefésüléses rendezés (2/3)

- Összevonási lépés: rendezett futamok összefésülése

//feltéve, hogy **N < M**

minden R_i fájlhoz egy lap lefoglalása
// N lap lefoglalása

minden R_i-ből egy-egy P_i lap beolvasása

ismétlés

az N lap közül a (rendezés szerint) első rekord kiválasztása,
legyen ez a P_j lapon

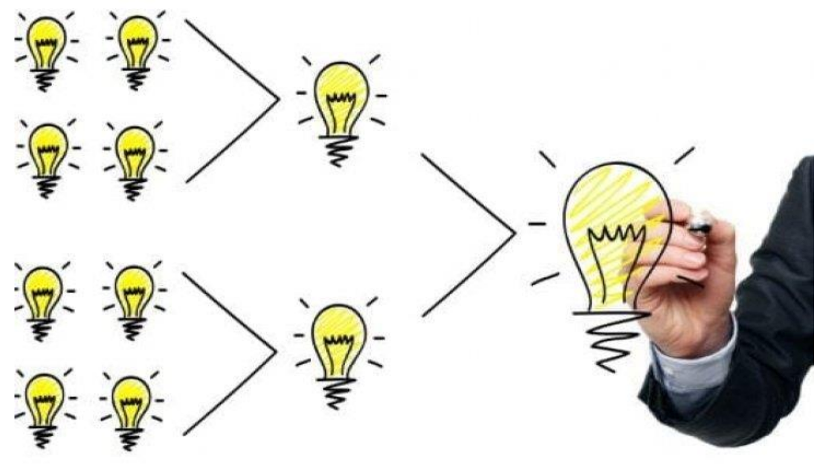
a rekord kiírása a kimenetre és törlése a P_j lapról

ha üres a lap, a következő P_j' beolvasása R_j-ből

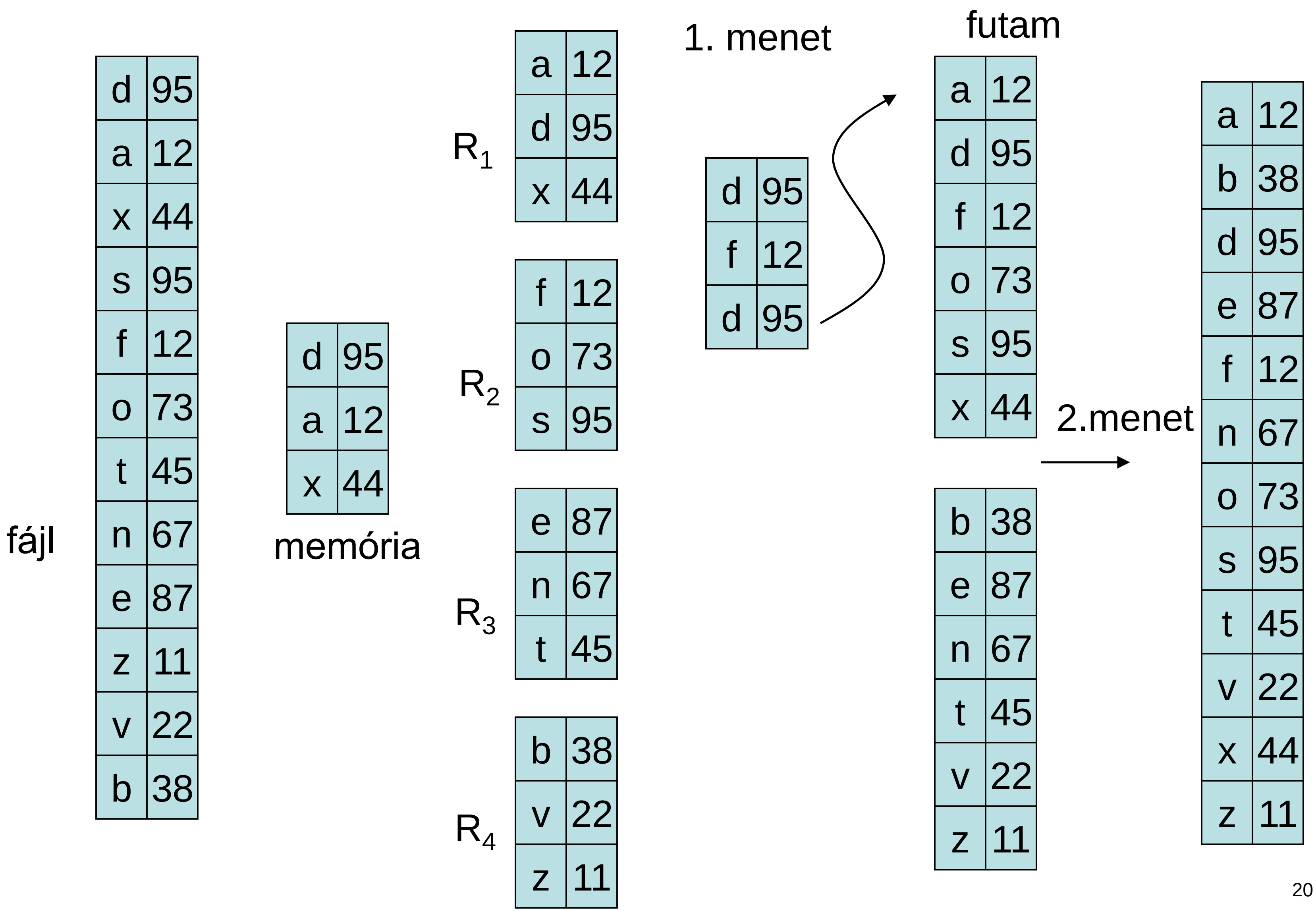
amíg minden lap ki nem ürül

Külső összefésüléses rendezés (3/3)

- Összevonási lépés: rendezett futamok összefésülése
- Mi van, ha **N > M** ?
  - több *menet*
  - minden *menet* M-1 futamot von össze, amíg nincs feldolgozva a reláció
  - a következő menetben a futamok száma kisebb és az összevonást ismételjük rekurzívan
  - a végső *menet*ben keletkezik a végső kimenet



Összefésüléses rendezés - példa



Összefésüléses rendezés költsége

- B_R: R lapjainak száma
- Rendezési lépés (rendezett futamok előállítása): 2 * B_R
  - reláció olvasása/írása
- Összevonási lépés (rendezett futamból M-1 darab összefésülése):
  - Hány futamot hajtunk végre?
  - kezdetben $\left\lceil \frac{B_R}{M} \right\rceil$ összevonandó futam
  - minden *menet* M-1 futamot rendez, vagyis a futamok számát mindig M-1-gyel kell osztani
  - tehát az összes menet száma: $\left\lceil \log_{M-1}\left(\frac{B_R}{M}\right) \right\rceil$
  - minden menetben 2 * B_R lapot olvasunk
    - reláció olvasása/írása
    - kivéve az utolsó kiírást
- Teljes számolási költség (az utolsó lépés, vagyis a rendezett tábla kiírása nélkül)
  - 2 * B_R + 2 * B_R * $\left\lceil \log_{M-1}\left(\frac{B_R}{M}\right) \right\rceil$ - B_R

Vetítés

- $\pi_{A1,A2\ldots}$ (R)
- felesleges mezők törlése
  - Végigolvassuk a táblát és a felesleges mezőket elhagyjuk, így kapjuk az R_1 táblát
  - Műveletigény: $B_R + B_{R1}$
- duplikált rekordok törlése
  - az R1 rendezése az összes mező alapján
  - Műveletigény kiírással együtt: (például külső összefésülés esetén):
  - $2 * B_{R1} + 2 * B_{R1} * \left\lceil \log_{M-1}\left(\frac{B_{R1}}{M}\right) \right\rceil$
  - a rendezett eredményt egyszer végigolvassuk, duplikált (szomszédos) rekordokat töröljük, így kapjuk az R2 táblát, azaz végeredményt
  - Műveletigény: $B_{R1} + B_{R2}$
- Teljes számítási költség
  - $B_R + B_{R1} + 2 * B_{R1} + 2 * B_{R1} * \left\lceil \log_{M-1}\left(\frac{B_{R1}}{M}\right) \right\rceil + B_{R1} + B_{R2}$

Felső becslés: $6 * B_R + 2 * B_R * \left\lceil \log_{M-1}\left(\frac{B_R}{M}\right) \right\rceil$

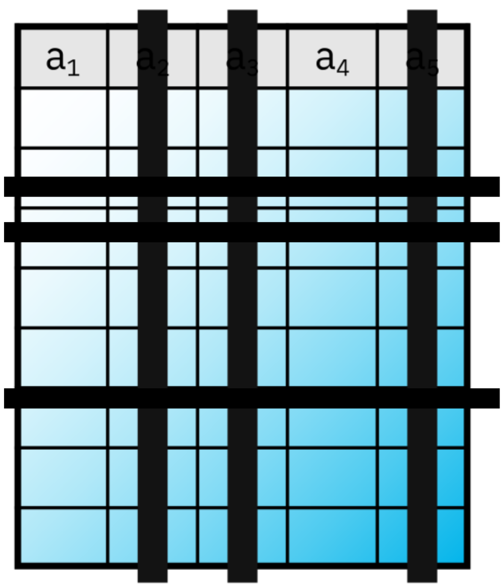
Méretbecslés - Vetítés

- $S := \pi_{A1,A2\ldots Ak}(R)$
- Felső becslés: B_R
- Ha csak egy A oszlopra vetítünk, akkor **V(A,R)** sor van a vetületben
  - Sorok száma: V(A,R) Blokkok száma: $V(A,R)/F_S$
- Több oszlop esetén különböző értékekből

  kapható különböző sorok maximális száma:
  - V(A1,R) * V(A2,R) * … * V(Ak,R)
  - A vetületben nem lehet több sor mint a táblában:[1]
  - Sorok száma:

    $\text{Min}(V(A1,R) * V(A2,R) * \ldots * V(Ak,R), N_R)$
  - Blokkok száma:

    $\text{Min}(V(A1,R) * V(A2,R) * \ldots * V(Ak,R), N_R) / F_S$



Unió

- P:= R ∪ S
- P esetén felső becslés: sorok száma: N_R+N_S blokkok száma: B_R+B_S
- duplikált rekordok törlése
  - az P rendezése az összes mező alapján
  - Műveletigény szerint kiírással együtt: (például külső összefésülés esetén):
  - $2 * B_P + 2 * B_P * \ | \log_{M-1}(B_P/M) \ |$
  - a rendezett eredményt egyszer végigolvassuk, duplikált (szomszédos) rekordokat töröljük, így kapjuk az P1 táblát, azaz végeredményt
  - Műveletigény: $B_P + B_{P1}$
- Teljes számítási költség
  - $2 * B_P + 2 * B_P * \ | \log_{M-1}(B_P/M) \ | + B_P + B_{P1}$

Felső becslés:

$4 * (B_R+B_S) + 2 * (B_R+B_S) * \ | \log_{M-1}((B_R+B_S)/M) \ |$

Különbség, Metszet, Szorzat visszavezetése az összekapcsolásra

- P:= R X S (= R ⋈ S speciális esete, ahol az ***R*** és ***S*** sémáknak nincs közös attribútuma)

  sorok száma: N_R X N_S blokkok száma: $B_R * N_S + B_S * N_R$
- P:= R ∩ S (= R ⋈ S speciális esete, ahol az ***R*** és ***S*** sémák teljesen megegyeznek)

  sorok száma: min(N_R , N_S) blokkok száma: min(B_R , B_S)
- P:= R - S

  sorok száma: N_R blokkok száma: B_R

A kiszámítást visszavezetjük a rendezéses-összefésüléses összekapcsolás kiszámítására, mivel az összes (t_R,t_S) sorpárnak egyszer be kell kerülni a memóriába, ahogy az összekapcsolások esetén is. (2 blokkot nyitunk a rendezett R és rendezett S számára. Ha R-beli sorhoz találunk S-beli sort, akkor kihagyjuk. Ha feldolgoztunk egy blokkpárt, akkor annak a helyére tölzünk be új blokkot, amelyiknek kisebb volt a legnagyobb értéke, mint a másik blok legnagyobb értéke.

Összekapcsolás

- $\pi_{name}(\sigma_{coursename=Advanced\ DBs}((student \bowtie_{cid} takes) \bowtie_{courseid} course))$
- megvalósítások
  - skatulyázott ciklusos összekapcsolás (nested loop join)
  - blokk-skatulyázott ciklusos összekapcsolás (block-nested loop join)
  - indexelt skatulyázott ciklusos összekapcsolás (index nested loop join)
  - összefésüléses rendező összekapcsolás (sort-merge join)
  - hasításos összekapcsolás (hash join)

Skatulyázott ciklusos összekapcsolás(1/2)

- R ⋈ S

  R minden t_R rekordján

  S minden t_S rekordján

  ha (t_R t_S illeszkedik) $t_R \bowtie t_S$ kiírása

  vége

  vége

- Bármilyen összekapcsolási feltételnél működik
  - Ha nincs illesztési feltétel, akkor a direkt szorzatot adja vissza
- S belső reláció
- R külső reláció

Skatulyázott ciklusos összekapcsolás(2/2)

- Költség:
  - legjobb eset, ha a kisebb reláció elfér a memóriában
    - ezt használjuk belső relációnak
    - $B_R + B_S$
  - legrosszabb eset, ha mindkét relációból csak 1-1 lap fér bele a memóriába
    - S-t minden R-beli rekordnál végig kell olvasni
    - $N_R * B_s + B_R$

Blokk-skatulyázott ciklusos összekapcsolás (1/2)

R minden X_R lapján

    S minden X_S lapján

        X_R minden t_R rekordján

            X_S minden t_S rekordján

                ha (t_R t_S illeszkedik) $t_R \bowtie t_S$ kiírása

            vége

        vége

    vége

vége

Blokk-skatulyázott ciklusos összekapcsolás (2/2)

- Költség:
  - legjobb eset, ha a kisebb reláció elfér a memóriában
    - ezt használjuk belső relációnak
    - $B_R + B_S$
  - legrosszabb eset, ha mindkét relációból csak 1-1 lap fér bele a memóriába
    - S-t minden R-beli lapnál végig kell olvasni
    - $B_R * B_s + B_R$ az előző esetben nagyobb volt: $N_R * B_s + B_R$

Indexelt skatulyázott ciklusos összekapcsolás

- $R \bowtie S$
- Index a belső reláción (S) – lehetőleg klaszterindex
- a külső reláció (R) minden rekordjánál *keresünk* a belső reláció indexe alapján illeszkedő sorokat az S-ből.
- Költség:
  - $B_R + N_R * c$
    - c a belső relációból index szerinti kiválasztás költsége
    - c = indexelt keresési költség + találat mérete blokkokban
    - Klaszterindex esetén $c = HTi + \lceil SC(A,R)/F_S \rceil =$

      $= \log(\text{index méret}) + \lceil (N_S / V(A,S)) / F_S \rceil =$

      $= \log(\text{index méret}) + \lceil (B_S / V(A,S) \rceil \approx \lceil (B_S / V(A,S) \rceil$ mert a logaritmusos tag sokkal kisebb
  - $B_R + N_R * B_S / V(A,S)$ klaszterindex és egyenletességi feltétel esetén
  - a kevesebb rekordot tartalmazó reláció legyen a külső

Összefésüléses rendező összekapcsolás

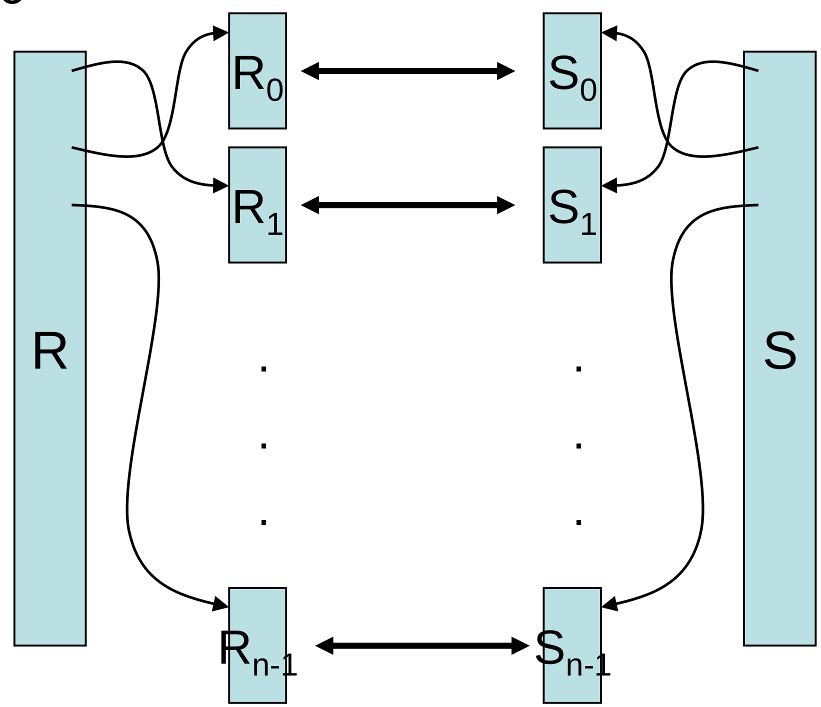
- R ⋈ S
- A relációk rendezettek az összekapcsolási mezők szerint
- Összefésüljük a rendezett relációkat
  - mutatók az első rekordra mindkét relációban
  - beolvasunk S-ből egy rekordcsoportot, ahol az összekapcsolási attribútum értéke megegyezik
  - beolvasunk rekordokat R-ből és feldolgozzuk
- A rendezett relációkat csak egyszer kell végigolvasni
- Költség:
  - rendezés költsége $+ B_S + B_R$

d	D
e	E
x	X
v	V

e	67
e	87
n	11
v	22
z	38

Hasításos összekapcsolás

- $R \bowtie S$
- R Is S-ben alkalmazzuk ugyanazt a *h hasító függvényt* az összekapcsolási mezőre és felosztjuk a rekordokat a memóriában elférő részekre (R_i és S_i férjen be a memóriába egyszerre)
  - R rekordjainak felosztása $R_0 \ldots R_{n-1}$
  - S rekordjainak felosztása $S_0 \ldots S_{n-1}$
- az egymáshoz illő, ugyanolyan indexű partíciók (kosarak) rekordjait összekapcsoljuk (mert ha két érték megegyezik, akkor a hasítófüggvény értékük is megegyezik, tehát csak azonos indexű partícióban lehetnek)
  - hasítófüggvény alapján indexelt blokk-skatulyázott ciklusos összekapcsolással
- Költség: $2*(B_R+B_S) + (B_R+B_S)$



Méretbecslés - összekapcsolás

- $R \bowtie S$
  - $R \cap S = \emptyset$ esetén $R \bowtie S = R \times S$

    sorok száma: $N_R * N_S$ blokkok száma: $B_R * N_S + B_S * N_R$
  - $R \cap S = \{A\}$, sem R-nek, sem S-nek nem kulcsa
    - $N_R * N_S / V(A,S)$ mert minden R-beli sorhoz $N_S / V(A,S)$ különböző érték illeszkedhet
    - $N_S * N_R / V(A,R)$ mert minden S-beli sorhoz $N_R / V(A,R)$ különböző érték illeszkedhet

    sorok száma: $N_S * N_R / \mathrm{Max}(V(A,R), V(A,S))$

    blokkok száma: $(B_R * N_S + B_S * N_R) / \mathrm{Max}(V(A,R), V(A,S))$

    Speciálisan például, ha $R.A \subseteq S.A$

    sorok száma: $N_S * N_R / V(A,S)$

    blokkok száma: $(B_R * N_S + B_S * N_R) / V(A,S)$
  - $R \cap S$ kulcs R-en esetén (S-nek idegen kulcsa)

    sorok száma: N_S blokkok száma: $(B_R * N_S + B_S * N_R) / N_R$

Összefoglalás

- Minden egyes művelet költségét és az eredmény méretét megadtuk
- A lekérdezési terv költségének becslését a költségek összegeként értelmeztük.
- Tovább lehetne javítani a költségeken materializálással, csövezetékesítés (pipeline), kommutatív, asszociatív egymás utáni műveletek optimális sorrendben történő elvégzésével, párhuzamos kiértékeléssel.

Köszönöm a figyelmet!