

# Relációs algebra 2.rész

Tankönyv: Ullman-Widom:  
Adatbázisrendszerek Alapvetés  
Második, átdolgozott kiadás,  
Panem, 2009



---

## 2.4. Egy algebrai lekérdező nyelv

- 01B\_RelAlg1alap: alpműveletek
  - 02A\_RelAlg2kif: lekérdező nyelv
- Tk. 2.4.1.Termék-feladatai a)-k)
- Tk. !! 2.4.10.feladat a hányadosra

# A relációs algebra alpműveletei ---1

**A hat alpművelet** (ismétlés, az 1.előadásról)

- $R \cup S$  unió
- $R - S$  különbség
- $\Pi_{\text{lista}} ( R )$  vetítés
- $\sigma_{\text{Feltétel}} ( R )$  kiválasztás
- $R \bowtie S$  természetes összekapcsolás
- $\rho_{S(B_1, \dots, B_k)} (R(A_1, \dots, A_k))$  átnevezés

Minimális készlet, vagyis bármelyiket elhagyva az a többivel nem fejezhető ki.

# A relációs algebra alpműveletei ---2

Alpműveletek egy másik készlete (5 alpművelet)  
(szorzás-jellegű műveletből választhatunk)

- $R \cup S$  unió
- $R - S$  különbség
- $\Pi_{\text{Lista}} ( R )$  vetítés
- $\sigma_{\text{Feltétel}} ( R )$  kiválasztás
- $R \times S$  természetes összekapcsolás

Hogyan lesz az algebrai műveletekből lekérdező nyelv? (megadjuk a felépítést és kiértékelést)

# Relációs algebrai lekérdező nyelv ---1

## Relációs algebrai kifejezés, mint lekérdező nyelv

Lekérdező nyelv: L -nyelv

Adott az adatbázis sémája:  $\mathbb{R} = \{R_1, \dots, R_k\}$

$q \in L$        $q: R_1, \dots, R_k \rightarrow V$  (eredmény-reláció)

E - relációs algebrai kifejezés:  $E(R_1, \dots, R_k) = V$  (output)

## Relációs algebrai kifejezések formális felépítése

### ■ Elemi kifejezések (alapkifejezések)

(i)  $R_i \in \mathbb{R}$  (az adatbázis-sémában levő relációnevek)

$R_i$  kiértékelése: az aktuális előfordulása

(ii) konstans reláció (véges sok, konstansból álló sor)

### ■ Összetett kifejezések (folyt. köv.oldalon)

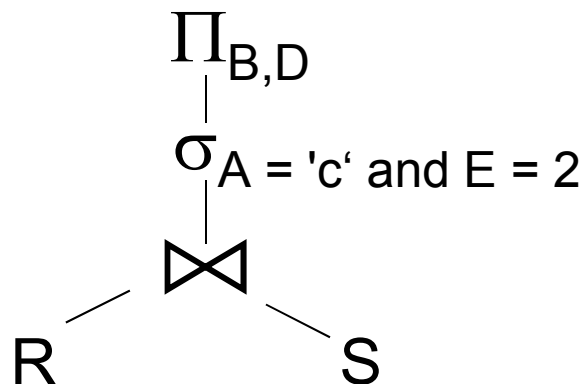
# Relációs algebrai lekérdező nyelv ---2

## (folyt.) Relációs algebrai kifejezések felépítése

- Összetett kifejezések
- Ha  $E_1, E_2$  kifejezések, akkor a következő  $E$  is kifejezés
  - $E := E_1 \cup E_2$  unió, ha azonos típusúak (és ez a típusa)
  - $E := E_1 - E_2$  különbség, ha  $E_1, E_2$  azonos típusúak (típus)
  - $E := \Pi_{\text{lista}} (E_1)$  vetítés (típus a lista szerint)
  - $E := \sigma_{\text{Feltétel}} (E_1)$  kiválasztás (típus nem változik)
  - $E := E_1 \bowtie E_2$  term. összekapcsolás (típus attr-ok uniója)
  - $E := \rho_{S(B_1, \dots, B_k)} (E_1 (A_1, \dots, A_k))$  átnevezés (típ.új attr.nevek)
  - $E := (E_1)$  kifejezést zárójelezve is kifejezést kapunk
- Ezek és csak ezek a kifejezések, amit így meg tudunk adni

# Lekérdezések kifejezése algebrában ---1

- Kifejezés kiértékelése: összetett kifejezést kívülről befelé haladva átírjuk kiértékelő fává, levelek: elemi kifejezések.
- A relációs algebra procedurális nyelv, vagyis nemcsak azt adjuk meg, hogy **mit** csináljunk, hanem azt is **hogyan**.
- Legyen R, S az R(A, B, C), S(C, D, E) séma feletti reláció  $\Pi_{B,D} \sigma_{A = 'c' \text{ and } E = 2} (R \bowtie S)$
- Ehhez **a kiértékelő fa**: (kiértékelése alulról felfelé történik)



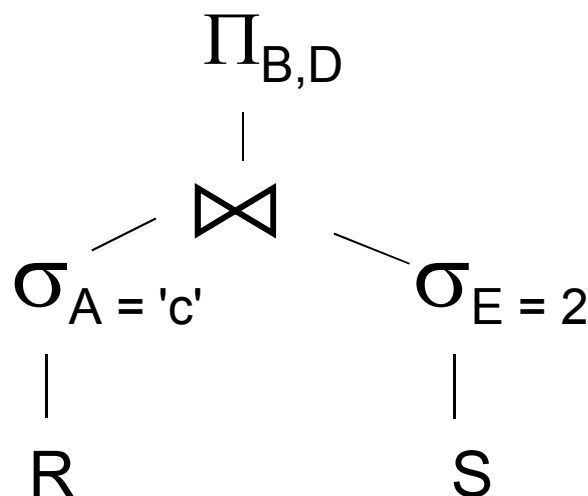
- Tudunk-e ennél jobb, hatékonyabb megoldást találni?

# Lekérdezések kifejezése algebrában ---2

- **Ekvivalens átalakítási lehetőségekkel**, relációs algebrai azonosságokkal át tudjuk alakítani a fentivel ekvivalens másik relációs algebrai kifejezésre. Hatékonyabb-e?

$$\Pi_{B,D} (\sigma_{A='c'}(R) \bowtie \sigma_{E=2}(S))$$

- Ehhez is felrajzolva a **kiértékelő fát**:



# Lekérdezések kifejezése algebrában ---3

- **Ekvivalens átalakítás:** oly módon alakítjuk át a kifejezést, hogy az adatbázis minden lehetséges előfordulására (vagyis ugyanazokra a táblákra) minden esetben ugyanazt az eredményt (ugyanazt a táblát) adja a két kiértékelő fa.
- **Adatbázisok-2 tárgyból** lesznek az **ekvivalens átalakítási szabályok**, a **szabály alapú optimalizálás** első szabálya például, hogy a kiválasztási műveletet minél előbb kell végrehajtani (közbülső táblák lehetőleg kicsik legyenek)
- Egy-egy részkifejezést, ha gyakran használjuk, akkor új változóval láthatjuk el, **segédváltozót vezethetünk be:**  
$$T(C_1, \dots, C_n) := E(A_1, \dots, A_n),$$
 de a legvégén a bevezetett változók helyére be kell másolni a részkifejezést.

# Példák relációs algebrai lekérdezésekre ---1

- Relációs algebra kifejezések ilyen bevezetése valóban használható a lekérdezések megadására? Tk.2.4.1.fela: Előző heti feladatokat megnéztük szemléletesen, most ugyanazokat formálisan relációs algebrai kifejezésként.

- **Példa:** Adottak az alábbi relációs sémák feletti relációk

Termék (gyártó, modell, típus)

PC (modell, sebesség, memória, merevlemez, cd, ár)

Laptop (modell, sebesség, memória, merevlemez, képernyő, ár)

Nyomtató (modell, színes, típus, ár)

- Jelölje:  $T(gy, m, t)$

$PC(m, s, me, ml, ár)$

$L(m, s, me, ml, k, ár)$

$Ny(m, sz, t, ár)$

Megj.: a két típus attr.név nem ugyanazt fejezi ki és így  $T \bowtie Ny$  természetes összekapcsolásnál „zűr”

## Példák relációs algebrai lekérdezésekre ---2

a.) Melyek azok a PC modellek, amelyek sebessége legalább 3.00?

$$\Pi_m(\sigma_{s \geq 3.00}(\mathbf{PC}))$$

b.) Mely gyártók készítenek legalább egy gigabájt méretű merevlemezzel rendelkező laptopot?

$$\Pi_{gy}(\sigma_{m \geq 100}(\mathbf{T} \bowtie \mathbf{L})) \text{ vagy ekv. } \Pi_{gy}(\mathbf{T} \bowtie (\sigma_{m \geq 100}(\mathbf{L})))$$

c.) Adjuk meg a B gyártó által gyártott összes termék modellszámát és árát típustól függetlenül!

**három részből áll (Nyomtató táblánál vigyázni, uis term.összekapcsolásnál a típus attr. itt mást jelent!)**

-- segédváltozót vezetek be, legyen  $\mathbf{BT} := \sigma_{gy='B'}(\mathbf{T})$

$$\begin{aligned} &\Pi_{m, ár}(\mathbf{BT} \bowtie \mathbf{PC}) \cup \Pi_{m, ár}(\mathbf{BT} \bowtie \mathbf{Laptop}) \cup \\ &\quad \cup \Pi_{m, ár}(\mathbf{BT} \bowtie \Pi_{m, ár}(\mathbf{Ny})) \end{aligned}$$

# Példák relációs algebrai lekérdezésekre ---3

d.) Adjuk meg valamennyi színes lézernyomtató

modellszámát:  $\Pi_m(\sigma_{sz='i'}(Ny)) \cap \Pi_m(\sigma_{t='lézer'}(Ny))$

-- elvégezhető más módon is:  $\Pi_m(\sigma_{sz='i' \wedge t='lézer'}(Ny)) =$   
 $= \Pi_m(\sigma_{sz='i'} \sigma_{t='lézer'}(Ny)) = \Pi_m(\sigma_{t='lézer'} \sigma_{sz='i'}(Ny))$

e) Melyek azok a gyártók, amelyek laptopot árulnak,  
PC-t viszont nem? (ha laptop gyártó több pc-t gyárt, akkor  
az eredménytábla csökken, **nem monoton** művelet: **R - S**)

$\Pi_{gy}(T \bowtie L) - \Pi_{gy}(T \bowtie PC)$

! f) Melyek azok a merevlemez méretek, amelyek legalább  
két PC-ben megtalálhatók? (táblát önmagával szorozzuk)

-- segédváltozót vezetek be, legyen **PC<sub>1</sub> := PC**

$\Pi_{PC.ml}(\sigma_{PC_1.m \neq PC.m \wedge PC_1.ml = PC.ml}(PC_1 \times PC))$

# Példák relációs algebrai lekérdezésekre ---4

! g) Adjuk meg azokat a PC-modell párokat, amelyek ugyanolyan gyorsak és a memóriájuk is ugyanakkora. Egy pár csak egyszer jelenjen meg, azaz ha már szerepel az (i, j), akkor a (j, i) ne jelenjen meg.

$$\Pi_{PC_1.m, PC.m}(\sigma_{PC_1.m < PC.m \wedge PC_1.s = PC.s \wedge PC_1.me = PC.me} (PC_1 \times PC))$$

!! h) Melyek azok a gyártók, amelyek gyártanak legalább két, egymástól különböző, legalább 2.80 gigahertzen működő számítógépet (PC-t vagy laptopot)

-- segédváltozó: **Gyors** :=  $\Pi_m(\sigma_{s \geq 2.8}(PC)) \cup \Pi_m(\sigma_{s \geq 2.8}(L))$

-- és ezzel legyen: **T<sub>1</sub>** := **T** ⋈ **Gyors** és **T<sub>2</sub>** := **T** ⋈ **Gyors**

$$\Pi_{T_1.gy}(\sigma_{T_1.gy = T_2.gy \wedge T_1.m \neq T_2.m} (T_1 \times T_2))$$

# Példák relációs algebrai lekérdezésekre ---5

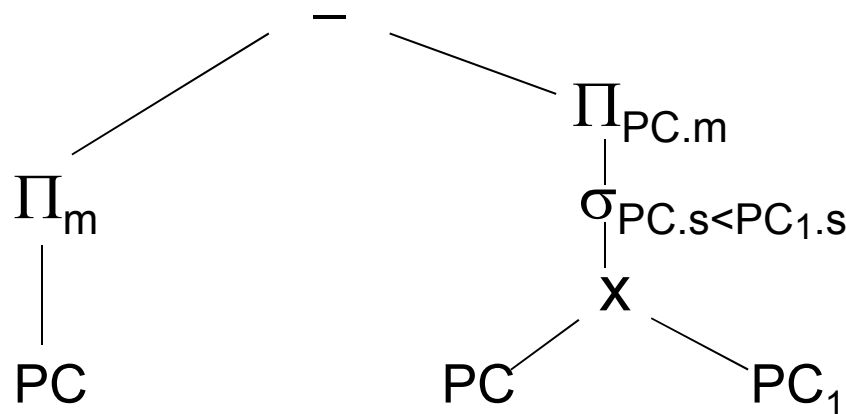
!! i) Melyik gyártó gyártja a leggyorsabb számítógépet (PC-t vagy laptopot)? **Lásd még az „elhagyás” típusú lekérdezéseket (köv.oldalon pl. maximum kifejezése)**

Kiválasztjuk azokat a PC-eket, amelyeknél van gyorsabb, ha ezt kivonjuk a PC-ékből megkapjuk a leggyorsabbat:

**EnnélVanNagyobb** =  $\Pi_{PC.m}(\sigma_{PC.s < PC_1.s}(PC \times PC_1))$

**Leggyorsabb**:  $\Pi_m(PC) - \text{EnnélVanNagyobb}$

Ehhez rajzoljuk fel a kiértékelő fát is: **(folyt.: PC helyett számítógép kell és a válaszban is a gyártó kell...)**



# Példák relációs algebrai lekérdezésekre ---6

!! j) Melyik gyártó gyárt legalább három, különböző sebességű PC-t? **mint a legalább kettő, csak ott 2x, itt 3x kell a táblát önmagával szorozni.** Legyenek  $S, S_1, S_2 := T \bowtie \Pi_{m,s}(PC)$

$\Pi_{S.gy}(\sigma_{S_1.gy=S.gy \wedge S_2.gy=S.gy \wedge S_1.s \neq S.s \wedge S_2.s \neq S.s \wedge S_1.s \neq S_2.s} (S \times S_1 \times S_2))$

!! k) Melyek azok a gyártók, amelyek pontosan három típusú PC-t forgalmaznak? **legalább 3-ból - legalább 4-t kivonni**

---

## ■ Mire érdemes felhívni a figyelmet?

Mi a leggyakrabban előforduló típus, amiből építkezek?

$\Pi_{lista}(\sigma_{feltétel}(\text{táblák szorzata}))$

Ezt a komponenst támogatja legerősebben majd az SQL:

**SELECT s-lista FROM f-lista WHERE feltétel;**

# „Elhagyás” típusú feladatok (különbséggel)

- Egy érdekes feladatot próbáljunk megoldani. Nézzük meg relációs algebrai kifejezéssel hogyan lehet egy attribútum **maximális értékét kiválasztani**.
- Legyen  $R(A,B)$  séma és adjuk meg  $A$  maximális értékét!
- Relációs algebra alapváltozatában erre nem vezettek be új műveletet (az SQL-ben majd összesítő függvényekkel). Hogyan tudjuk megállapítani, hogy egy érték maximális-e? Úgy, hogy melléteszünk egy másik attribútumértéket és ha az nagyobb, akkor ez nem lehet maximális, vagyis hibás  
$$\text{Hibás} = \sigma_{R1.A < R2.A} (R_1 \times R_2)$$
$$\text{JóVálasz} = \text{Összes} - \text{Hibás} = \pi_A(R) - \pi_A(\text{Hibás})$$
- Egy másik „elhagyás” típusú feladat, Tk. 2.4.10.feladata: **a hányados** (Innen a további rész már csak szorgalmi)

# Hányados, osztás művelete ---1

- Maradékos osztás:  $7 \div 3 = 2$ , mert 2 a legnagyobb egész, amelyre még  $2 * 3 \leq 7$ . Ennek a mintájára:
- Relációk szorzata esetén  $\leq$  helyett tartalmazás.
- R és S sémája  $R(A_1, \dots, A_n, B_1, \dots, B_m)$ , illetve  $S(B_1, \dots, B_m)$ ,  
 $Q = R \div S$  sémája  $Q(A_1, \dots, A_n)$
- $R \div S$  a legnagyobb (**legtöbb sort tartalmazó**) reláció, amelyre  $(R \div S) \times S \subseteq R$ .
- Az R reláció sémája  $(A_1, \dots, A_n, B_1, \dots, B_m)$  és az S reláció sémája  $(B_1, \dots, B_m)$ , azaz S összes attribútuma benne van R attribútumainak halmazában. Az R és S hányadosa  $R \div S$  megadja azon  $A_1, \dots, A_n$  attribútumú t sorok halmazát, amelyekre igaz, hogy az S reláció minden s sorára a ts sor benne van az R relációban! Ez kifejezhető rel.algebrában!

# Hányados, osztás művelete ---2

Ki szereti legalább azokat a gyümölcsöket, mint Micimackó?

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

| MIT   |
|-------|
| málna |
| méz   |

 $\div$ 

| KI        |
|-----------|
| Füles     |
| Micimackó |

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

## Hányados, osztás művelete ---3

$$R(A,B) \div S(B) = \Pi_{A_1, \dots, A_n}(R) - \Pi_{A_1, \dots, A_n}(\Pi_{A_1, \dots, A_n}(R) \times S - R)$$

- Képezzük az összes lehetséges sorokat

$$E1 = \Pi_{A_1, \dots, A_n}(R) \times S$$

- Mit kell leellenőrizni? Mely sorok nincsenek ebben?

$$E2 = E1 - R = \Pi_{A_1, \dots, A_n}(R) \times S - R$$

- Ebben olyan  $t$  sorok szerepelnek, amelyek nem jók, ennek vesszük A-ra a vetületét – ezek a rosszak

$$E3 = \Pi_{A_1, \dots, A_n}(E2) = \Pi_{A_1, \dots, A_n}(\Pi_{A_1, \dots, A_n}(R) \times S - R)$$

- A jókat megkapjuk, ha az összesből kivonjuk a rosszakat

$$\text{össz} - E3 = \Pi_{A_1, \dots, A_n}(R) - \Pi_{A_1, \dots, A_n}(\Pi_{A_1, \dots, A_n}(R) \times S - R)$$

- Tehát a hányados kifejezhető relációs algebrában:

$$\Pi_{A_1, \dots, A_n}(R) - \Pi_{A_1, \dots, A_n}(\Pi_{A_1, \dots, A_n}(R) \times S - R)$$

# PÉLDA (angol nyelvű) Ullman könyv példája

## Sörivók adatbázisséma (köv.lapon magyarul)

Beers(name, manf)

Bars(name, addr, license)

Drinkers(name, addr, phone)

Likes(drinker, beer)

Sells(bar, beer, price)

Frequents(drinker, bar)

- Az aláhúzás jelöli a **kulcsot** (a sorok a kulcs összes attribútumán nem vehetik fel ugyanazt az értékeket).
  - Ez a kulcs, külső kulcs és hivatkozási épség megszorításoknak lesz később kiváló példája, magyar fordításban „name” helyett: sör, bár, név.

Példák relációs algebrai lekérdezésekre  
a sörivók adatbázison, amelynek sémája

Sörök(név, gyártó)

Bárok(név, város, engedély)

Ivók(név, város, tel)

Kedvel(név, sör)

Felszolgál(bár, sör, ár)

Látogat(név, bár)

- **Feladatok:** Hogyan definiálnánk a boldog ivót?
- Boldog ivó = látogat olyan bárt, ahol felszolgálnak olyan sört, amit kedvel. Fejezzük ki relációs algebrában!

# Szorgalmi feladatok lekérdezésekre ---1

$K(N,S)$  jelölje  $Kedvel(\underline{név}, \underline{sör})$  (név jelölje a sörivót)

$F(B,S)$   $Felszolgál(\underline{bár}, \underline{sör})$  (most „ár” nélkül!)

$L(N,B)$   $Látogat(\underline{név}, \underline{bár})$

- **Boldog ivó** = látogat olyan bárt, ahol felszolgálnak olyan sört, amit kedvel. Fejezzük ki relációs algebrában!
- $(N,S,B)$  hármások: ahol  $N$  szereti  $S$ -t és  $N$  jár  $B$ -be:  
 $nsb := K(N,S) \bowtie L(N,B)$  (Megj.:  $\bowtie$  kommutatív)
- $(N,S,B)$  hármások: ahol  $N$  szereti  $S$ -t, és  $N$  jár  $B$ -be és felszolgálnak  $S$ -t a  $B$ -ben:  
 $h := (K(N,S) \bowtie L(N,B)) \bowtie F(B,S)$  (Megj.:  $\bowtie$  asszociatív)
- **Megoldás**:  $Boldog\_ivók := \Pi_N(h)$

# Szorgalmi feladatok lekérdezésekre ---2

- Kik a Boldogtalan\_ivók?  $\Pi_N(K)$  – Boldog\_ivó
- Ki jár olyan bárba, ahol van legalább két kedvenc söre?
- Ki jár CSAK olyan bárba, ahol legalább egy kedvenc söre kapható? (MÉG BOLDOGABB)
- Ki jár olyan bárba, ahol az összes kedvenc söre kapható? (NAGYON BOLDOG)
- Ki jár CSAK olyan bárba, ahol az összes kedvenc söre kapható? (SZUPER BOLDOG)
- Ki jár olyan bárba, ahol mindent szeret?
- Ki jár CSAK olyan bárba, ahol mindent szeret?
- „Tanácsadó szolgálat”: Hova menjen el két ivó sörözni, olyan bárt keresünk, ahova mind a ketten járnak és ahol mind a ketten találnak olyan sört, amit kedvelnek...stb...

# Köv.téma: Lekérdezések SQL-ben

- Hogyan tudjuk a fenti feladatokat és példákat, az itt szereplő lekérdezéseket kifejezni gépes gyakorlati környezetben az SQL-ben?
- Relációs algebrai kifejezések átírása az SQL SELECT utasításra: 04A\_RelAlg3ToSQL