

# Alkérdeések az SQL SELECT-ben

## Relációs algebrai lekérdezések (példák)

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

---

### 6.3. Alkérdeések (SQL SELECT)

Folyt.2.4. Relációs algebra, mint  
lekérdező nyelv

Példák: Tk. Termékek feladatai



## (2EA) ismétlés: SFW alapértelmezése

- Tk.6.2.fej.: 2.EA: Több táblára vonatkozó lekérdezések:

SELECT [DISTINCT] kif<sub>1</sub> [[AS] onév<sub>1</sub>], ..., kif<sub>n</sub> [[AS] onév<sub>n</sub>]

FROM R<sub>1</sub> [t<sub>1</sub>], ..., R<sub>n</sub> [t<sub>n</sub>]

WHERE feltétel (vagyis logikai kifejezés)

Alapértelmezés (a műveletek szemantikája -- általában)

- A FROM záradékban levő relációkhoz tekintünk egy-egy **sorváltozót**, amelyek a megfelelő reláció minden sorát bejárják (beágyazott ciklusban)
- Minden egyes „aktuális” sorhoz kiértékeljük a WHERE záradékot (csak az igaz sorok kerülnek az eredménybe)
- A SELECT záradékban szereplő kifejezéseknek megfelelően képezzük a sorokat. Ha van DISTINCT, akkor az ismétlődő sorokat elhagyjuk.

# (2EA) ismétlés: Halmazműveletek az SQL-ben

- A relációs algebrai halmazműveletek: **unió**, **különbség** és **metszet**, ebből csak az unió és különbség alapművelet, az SQL-ben mindhárom használható, implementálva van
- A **SELECT-FROM-WHERE** utasítások általában **multihalmaz** szemantikát használnak, külön kell kérni **DISTINCT**-tel ha halmazt szeretnénk kapni, viszont a **halmazműveleteknél** mégis a **halmaz szemantika** az érvényes, itt a **multihalmaz szemantikát** kell kérni: **ALL**
- Az SQL-ben a halmazműveleteket úgy vezették be, hogy azt mindig két lekérdezés között lehet értelmezni:

(SFW-lekérdezés1)

[ **UNION** [ALL] |  
INTERSECT [ALL] |  
{EXCEPT | MINUS} [ALL] ]

(SFW-lekérdezés2);

# Alkérdeések

- Zárójelezett SFW SELECT-FROM-WHERE utasításokat **(alkérdeéseket)** is használhatunk a **WHERE** záradékban, **HAVING** záradékban (később lesz) és a **FROM** listán is.
- **Szintaktikus alakja:** zárójelbe kell tenni a lekérdezést
- Hol használható? Ott, ahol relációnevet használunk:

(1) **WHERE és HAVING záradékban:** kifejezésekben, feltételekben

(2) **FROM listában:** új listaelem (rel.név változó SQL-ben)  
**(lekérdezés) [AS] sorváltozó**

Ez felel meg annak, ahogyan a relációs algebrában tetsz.helyen használhattuk a lekérdezés eredményét.

# Alkérdeések a WHERE záradékban

## WHERE záradékban:

- (i) Az alkérdés eredménye egyetlen **skalárérték**, vagyis az alkérdés olyan, mint a konstans, ami egy új elemi kifejezésként tetszőleges kifejezésben használható.
- (ii) **Skalár értékekből álló multihalmaz** logikai kifejezésekben használható:
  - [NOT] EXISTS (lekérdezés)
  - kifejezés [NOT] IN (lekérdezés)
  - kifejezés  $\Theta$  [ANY | ALL] (lekérdezés)
- (iii) **Teljes, többdimenziós tábla** a visszatérő érték:
  - [NOT] EXISTS (lekérdezés)
  - (kif<sub>1</sub>, ... kif<sub>n</sub>) [NOT] IN (lekérdezés)

# Alkérdések a WHERE záradékban

- Milyen változók szerepelhetnek egy alkérdésben?
  - Lokális saját változói a saját FROM listáról
  - Külső kérdés változói: ekkor az alkérdés korrelált.

## Szemantikája

- Ha az alkérdés **nem korrelált**, önállóan kiértékelhető és ez az eredmény a külső kérdés közben nem változik, a külső kérdés szempontjából ez egy konstanstábla, akkor a kiértékelés mindig a legbelsőből halad kifelé.
- **Korrelált alkérdés**, amely többször kerül kiértékelésre, minden egyes kiértékelés megfelel egy olyan értékadásnak, amely az alkérdésen kívüli sorváltozóból származik (ezt később, példákon keresztül mutatjuk be)

# Skalár értéket visszaadó alkérdések

- Ha egy alkérdés biztosan egy attribútumon egy sort ad vissza eredményként (egyelemű), akkor úgy használható, mint egy konstans érték.
  - az eredmény sornak egyetlen oszlopa van.
  - Futásidejű hiba keletkezik, ha az eredmény nem tartalmaz sort, vagy több sort tartalmaz.
- **Példa: Felszolgál(bár, sör, ár)** táblában keressük meg azokat a bárokat, ahol a Miller ugyanannyiba kerül, mint Joe bárjában a Bud.
- Két lekérdezésre biztos szükségünk lesz:
  1. Mennyit kér Joe a Budért?
  2. Melyik kocsmákban adják ugyanennyiért a Millert?

# Skalár értéket visszaadó alkérdések

**Példa:** Felszolgal(bár, sör, ár) táblában keressük meg azokat a bárokat, ahol a Miller ugyanannyiba kerül, mint Joe bárjában a Bud.

```
SELECT bár
```

```
FROM Felszolgal
```

```
WHERE sör = 'Miller' AND
```

```
    ár = (SELECT ár
```

```
        FROM Felszolgal
```

```
        WHERE bár = 'Joe' 's bar'
```

```
        AND sör = 'Bud' );
```

Ennyit kér  
Joe a Budért.



# Tk.példa: Skalár értéket adó alkérdések

- Csillagok háborúja film gyártásirányítója:

```
SELECT név
```

```
FROM GyártásIrányító
```

```
WHERE azonosító =
```

```
  (SELECT producerAzon
```

```
    FROM Filmek
```

```
    WHERE cím = 'Csillagok háborúja'
```

```
  );
```

# Skalár értékekből álló multihalmazt visszaadó alkérdések: ANY művelet

- $x = \text{ANY}(\text{alkérdés})$  akkor és csak akkor igaz, ha  $x$  egyenlő az alkérdés legalább egy sorával.  
= helyett bármilyen aritmetikai összehasonlítás szerepelhet.
- **Példa:**  $x > \text{ANY}(\text{alkérdés})$  akkor igaz, ha  $x$  nem az alkérdés legkisebb elemével azonos.
  - Itt az alkérdés sorai egy mezőből állnak.

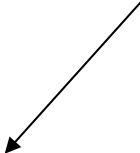
# Skalár értékekből álló multihalmazt visszaadó alkérdések: ALL művelet

- $x \leftrightarrow \text{ALL}(\text{alkérdés})$  akkor és csak akkor igaz, ha  $x$  az alkérdés egyetlen sorával sem egyezik meg.
- $\leftrightarrow$  helyett tetszőleges összehasonlítás szerepelhet.
- **Példa:**  $x \geq \text{ALL}(\text{alkérdés})$   $x$  az alkérdés eredményének maximum értékével azonos.

# Példa: ALL

```
SELECT sör
FROM Felszolgál
WHERE ár >= ALL(
    SELECT ár
    FROM Felszolgál);
```

A külső lekérdezés  
Felszolgáljának söre  
egyetlen alkérdésbeli  
sörnél sem lehet  
olcsóbb.



# Az IN művelet a WHERE záradékban

- sor IN (alkérdés) akkor és csak akkor **igaz**, ha a sor eleme az alkérdés eredményének (itt a sor egy sor/tuple, nem sör)
- Tagadás: sor NOT IN (alkérdés).
- Az IN-kifejezések a WHERE záradékban jelenhetnek meg
- Példa:

```
SELECT *
```

```
FROM Sörök
```

```
WHERE név IN (SELECT sör
```

A sörök,  
melyeket  
Fred szeret.

```
FROM Szeret
```

```
WHERE név = 'Fred' );
```

# Tk.példa: Sorokat tartalmazó feltételek

- Harrison Ford filmjeinek gyártásirányítója:

```
SELECT név
FROM GyártásIrányító
WHERE azonosító IN
    (SELECT producerAzon
     FROM Filmek
     WHERE (cím, év) IN
         (SELECT filmCím, filmév
          FROM SzerepelBenne
          WHERE színész = 'Harrison Ford' )
    );
```

# Mi a különbség?

```
SELECT a  
FROM R, S  
WHERE R.b = S.b;
```

```
SELECT a  
FROM R  
WHERE b IN (SELECT b FROM S);
```

# IN az R soraira vonatkozó predikátum

```
SELECT a  
FROM R  
WHERE b IN (SELECT b FROM S);
```

Egy ciklus R sorai  
fölött.

a	b
1	2
3	4

R

b	c
2	5
2	6

S

(1,2) kielégíti a  
feltételt;  
1 egyszer jelenik  
meg az  
eredményben.



# Itt R és S sorait párosítjuk

```
SELECT a
FROM R, S
WHERE R.b = S.b;
```

Dupla ciklus R és S  
sorai fölött

a	b
1	2
3	4

R

b	c
2	5
2	6

S

(1,2) és (2,5)  
(1,2) és (2,6)  
is kielégíti a  
feltételt;  
1 kétszer kerül  
be az eredménybe.

# Az EXISTS művelet a WHERE-ben

- EXISTS (alkérdés) akkor és csak akkor igaz, ha az alkérdés eredménye nem üres.
  - Tagadása: NOT EXISTS (alkérdés)
- Példa: A Sörök(név, gyártó) táblában keressük meg azokat a söröket, amelyeken kívül a gyártójuk nem gyárt másikat.
- Ez korrelált alkérdés, többször kerül kiértékelésre, a külső tábla minden sorára kiértékeljük az alkérdést.
- A korrelált lekérdezések használata közben figyelembe kell vennünk a nevek érvényességi körére vonatkozó szabályokat.

# Példa: EXISTS

```
SELECT név
FROM Sörök b1
WHERE NOT EXISTS
  (SELECT *
   FROM Sörök
   WHERE gyártó = b1.gyártó
    AND név <> b1.név);
```

Azon b1  
sörtől  
különböző  
sörök,  
melyeknek  
ugyanaz  
a gyártója.

Változók láthatósága: itt  
a gyártó a legközelebbi  
beágyazott FROM-beli  
Táblából való, aminek  
van ilyen attribútuma.

A „nem  
egyenlő”  
művelet  
SQL-ben.

# Tk.példa: Korrelált alkérdés

- A több, mint egyszer előforduló filmcímek megkeresése:

```
SELECT DISTINCT cím
FROM Filmek AS Régi
WHERE év < ANY
  (SELECT év
   FROM Filmek
   WHERE cím = Régi.cím
  );
```

# Alkérdeések a FROM záradékban

4.EA: ALKÉRDÉSEK WHERE feltételben

Folytatása következik:

5.EA: ALKÉRDÉSEK HAVING feltételben

6.EA: ALKÉRDÉSEK FROM záradékban

# Relációs algebrai lekérdezések (példák)

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

---

Példák: Relációs algebra és SQL  
(Tankönyv Termékek feladatai)



# (2ea) ismétlés: Relációs algebra ---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)

# (2ea) ismétlés: Relációs algebra ---2

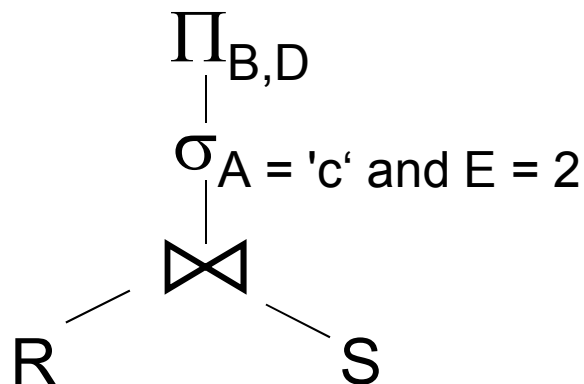
## (folyt.) Relációs algebrai kifejezések felépítése

- **Összetett kifejezések**
- Ha  $E_1, E_2$  kifejezések, akkor a következő  $E$  is kifejezés
  - $E := \Pi_{\text{lista}} ( E_1 )$  vetítés (típus a lista szerint)
  - $E := \sigma_{\text{Feltétel}} ( E_1 )$  kiválasztás (típus nem változik)
  - $E := E_1 \cup E_2$  unió, ha azonos típusúak (és ez a típusa)
  - $E := E_1 - E_2$  különbség, ha  $E_1, E_2$  azonos típusúak (típus)
  - $E := E_1 \bowtie E_2$  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)



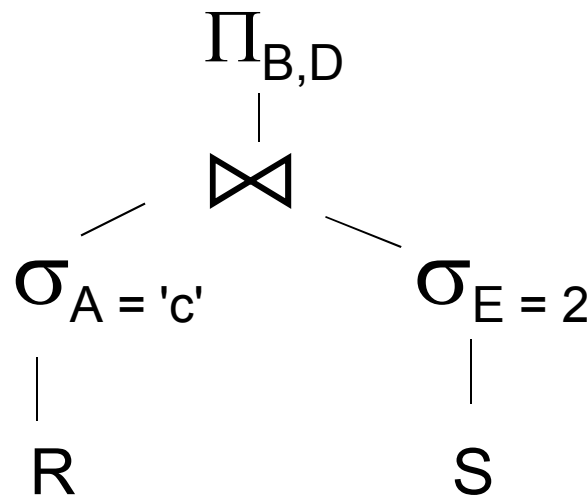
- Tudunk-e ennél jobb, hatékonyabb megoldást találni?

# 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 bármilyen is a táblák tartalma) minden esetben ugyanazt az eredményt (vagyis ugyanazt az output táblát) adja az eredeti és az átalakított 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)
- Ha 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.

# Feladatok ---1

- Először relációs algebrában táblákkal gondolkodva nézzük meg, hogy milyen műveletekkel tudjuk megkapni a kívánt eredményt, írjuk fel lineáris módon és kifejezőfákkal, majd a kifejezőfákat átírva SQL lekérdezésekre többféle megoldási lehetőséget vizsgáljunk meg, vessünk össze
- A mai előadáson: Tankönyv -- Termékek feladatai:
- [http://people.inf.elte.hu/sila/AB1ea/Feladatok\\_Termekek.pdf](http://people.inf.elte.hu/sila/AB1ea/Feladatok_Termekek.pdf)  
create table: [http://people.inf.elte.hu/sila/eduAB/create\\_termekek.txt](http://people.inf.elte.hu/sila/eduAB/create_termekek.txt)
- További feladatok: Tankönyv -- Csatahajós feladatai:
- [http://people.inf.elte.hu/sila/AB1ea/Feladatok\\_Csatahajok.pdf](http://people.inf.elte.hu/sila/AB1ea/Feladatok_Csatahajok.pdf)  
create table: [http://people.inf.elte.hu/sila/eduAB/create\\_csatahajok.txt](http://people.inf.elte.hu/sila/eduAB/create_csatahajok.txt)

# Feladatok ---2

Legyen adott az alábbi **relációs sémák** feletti relációk:

Termék (gyártó, modell, típus)

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

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

Nyomtató (modell, színes, típus, ár)

**Feladatok Tk.2.4.1.feladat** (ezeket a kérdéseket konkrét táblák alapján természetes módon meg lehet válaszolni, majd felírjuk relációs algebrában)

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

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

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

!! i) Melyik gyártó gyártja a leggyorsabb számítógépet (laptopot vagy PC-t)?

!! k) Melyek azok a gyártók, akik pontosan három típusú PC-t forgalmaznak?

# Példák relációs algebrai lekérdezésekre ---1

- Relációs algebra kifejezések ilyen bevezetése valóban használható a lekérdezések megadására?

- Tk.2.4.1.feladat

- **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:  $T(\text{gy}, \text{m}, \text{t})$

$PC(\text{m}, \text{s}, \text{me}, \text{ml}, \text{ár})$

$L(\text{m}, \text{s}, \text{me}, \text{ml}, \text{k}, \text{ár})$

$Ny(\text{m}, \text{sz}, \text{t}, \text{ár})$

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?

## Példák átírásokra ---2

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

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

SELECT modell  
FROM PC  
WHERE sebesség >= 3.00;

$\Pi_m$   
|  
 $\sigma_{s \geq 3.00}$   
|  
PC



## Példák átírásokra ---2

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

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

SELECT modell  
FROM PC  
WHERE sebesség >= 3.00;

$\Pi_m$   
|  
 $\sigma_{s \geq 3.00}$   
|  
PC

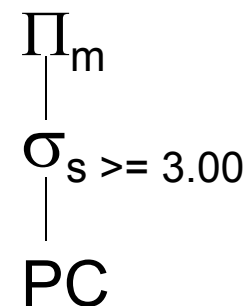
b.) Mely gyártók készítenek legalább száz gigabájt méretű merevlemezzel rendelkező laptopot?

## Példák átírásokra ---2

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

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

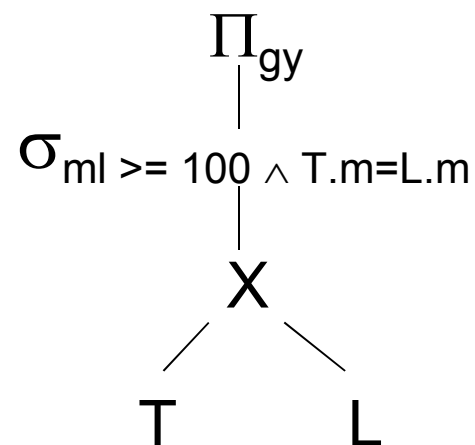
SELECT modell  
FROM PC  
WHERE sebesség >= 3.00;



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

$\Pi_{gy}(\sigma_{ml \geq 100}(T \bowtie L))$

SELECT gyarto  
FROM Termek T, Laptop L  
WHERE merevlemez >= 100  
AND T.modell=L.modell;



## Példák átírásokra ---2

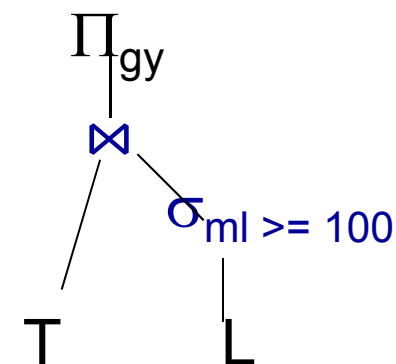
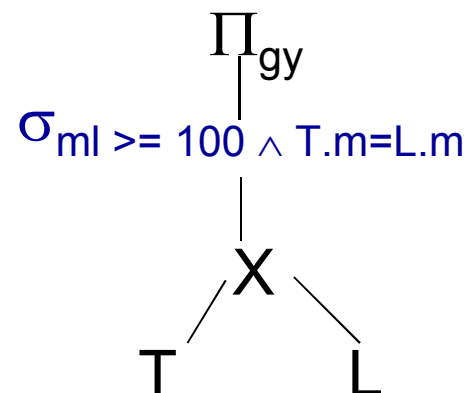
b.) Mely gyártók készítenek legalább száz gigabájt méretű merevlemezzel rendelkező laptopot?

$\Pi_{gy} (\sigma_{ml \geq 100} (T \bowtie L))$

SELECT gyarto  
FROM Termek natural join Laptop  
WHERE merevlemez >= 100

SELECT gyarto  
FROM Termek T, Laptop L  
WHERE merevlemez >= 100  
AND T.modell=L.modell;

SELECT gyarto FROM Termek  
WHERE modell IN  
(SELECT modell FROM PC  
WHERE merevlemez >= 100);



## 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 száz gigabájt méretű merevlemezzel rendelkező laptopot?

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

# 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 száz gigabájt méretű merevlemezzel rendelkező laptopot?

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

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

## 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 száz 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 más jelent!)**

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

## 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 száz 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  $\mathbf{BT} := \Pi_m \sigma_{gy='B'}(\mathbf{T})$

$$\Pi_{m, \text{ár}}(\mathbf{BT} \bowtie \mathbf{PC}) \cup \Pi_{m, \text{ár}}(\mathbf{BT} \bowtie \mathbf{Laptop}) \cup$$

$$\cup \Pi_{m, \text{ár}}(\mathbf{BT} \bowtie \mathbf{Ny})$$

## c.) SQL-ben kifejezve

```
select modell, ar from pc
where modell in
  (select modell from termek
   where gyarto='B')
```

union

```
select modell, ar from laptop
where modell in
  (select modell from termek
   where gyarto='B')
```

union

```
select modell, ar from nyomtato
where modell in
  (select modell from termek
   where gyarto='B');
```



c.) --- mint az előző, egyszerűbben,  
„with” még nem kell, visszatérünk

with

Btermek as

(select modell from termek where gyarto='B')

select modell, ar from pc natural join Btermek

union

select modell, ar from laptop natural join Btermek

union

select modell, ar from nyomtato natural join Btermek;

# Példák relációs algebrai lekérdezésekre ---3

d.) Adjuk meg valamennyi színes lézernyomtató modellszámát

## Példák relációs algebrai lekérdezésekre ---3

d.) Adjuk meg valamennyi színes lézernyomtató

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

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

$= \Pi_m(\sigma_{sz='i'} \sigma_{t='lézer'}(\mathbf{Ny})) = \Pi_m(\sigma_{t='lézer'} \sigma_{sz='i'}(\mathbf{Ny}))$

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**)

# Példák relációs algebrai lekérdezésekre ---3

d.) Adjuk meg valamennyi színes lézernyomtató

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

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

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**)

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

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

# Példák relációs algebrai lekérdezésekre ---3

d.) Adjuk meg valamennyi színes lézernyomtató

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

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

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**)

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

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

# Példák relációs algebrai lekérdezésekre ---3

d.) Adjuk meg valamennyi színes lézernyomtató

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

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

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**)

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

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

-- segédváltozót vezetek be, legyen  $\mathbf{PC}_1 := \mathbf{PC}$

$\Pi_{PC.ml}(\sigma_{PC_1.m \neq PC.m \wedge PC_1.ml = PC.ml}(\mathbf{PC}_1 \times \mathbf{PC}))$

# 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.

# 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)



# 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:  $\mathbf{Gyors} := \Pi_m(\sigma_{s \geq 2.8}(PC)) \cup \Pi_m(\sigma_{s \geq 2.8}(L))$

-- és ezzel legyen:  $\mathbf{T_1} := T \bowtie \mathbf{Gyors}$  és  $\mathbf{T_2} := T \bowtie \mathbf{Gyors}$

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

# Példák relációs algebrai lekérdezésekre ---5

!! i) Melyik gyártó gyártja a leggyorsabb PC-t?

(„elhagyás” típusú lekérdezések, nincs nála gyorsabb PC)

# Példák relációs algebrai lekérdezésekre ---5

!! i) Melyik gyártó gyártja a leggyorsabb PC-t?

(az „elhagyás” típusú lekérdezések, lásd maximum kif.)

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:

# 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:

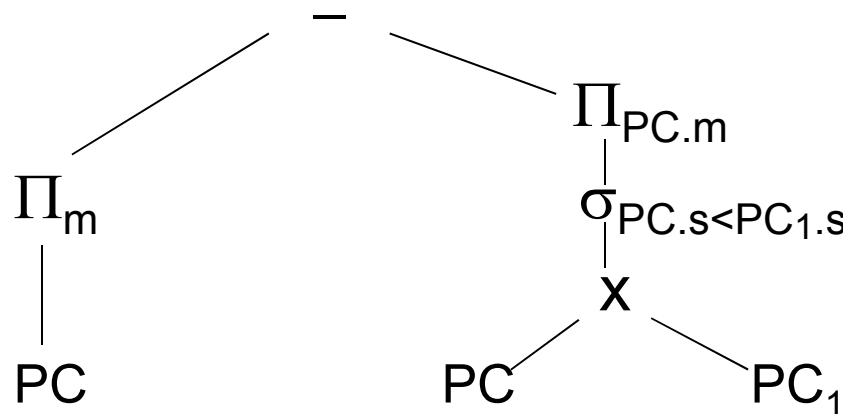
**EnnélVanNagyobb** =  $\Pi_{PC.m}(\sigma_{PC.s < PC_1.s}(PC \times PC_1))$

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

Ehhez rajzoljuk fel a kiértékelő fát is: (folyt.: PC helyett számítógép kell

és a válaszban

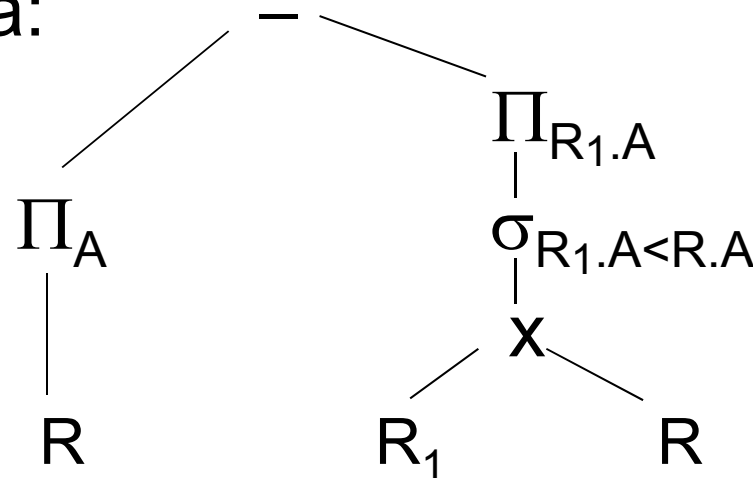
is a gyártó kell...)



# MAX előállítás relációs algebrában

- Nézzük meg a maximum előállításának a kérdését! Legyen  $R(A,B)$ . **Feladat:** Adjuk meg  $MAX(A)$  értékét! (Ez majd átvezet az új témára, aggregáló függvényekre, illetve csoportosításra).
- $\pi_A(R) - \pi_{R1.A}(\sigma_{R1.A < R.A}(\rho_{R1}(R) \times R))$

- Kiértékelő fa:



# Folyt. Rel.alg. kifejezés átírása SQL-re

- Előző oldal folyt.max előállítás átírása SQL-re:
- Kiértékelő fa szerinti átírás SQL-be:

```
(SELECT A FROM R)  
EXCEPT  
(SELECT R1.A AS A  
FROM R R1, R R2  
WHERE R1.A<R2.A);
```

- Nézzük meg korrelált (függő) alkérdéssel is:

```
SELECT A FROM R MAXA  
WHERE NOT EXISTS  
(SELECT A FROM R  
WHERE A > MAXA.A);
```

# 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?

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

Ezt a komponenst támogatja legerősebben majd az SQL:

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

# Kérdés / Válasz

- **Köszönöm a figyelmet! Kérdés/Válasz?**
- Először relációs algebrában táblákkal gondolkodva felírjuk kifejezőfákkal, majd átírva SQL lekérdezésekre többféle megoldási lehetőséget vizsgáljunk meg, vessünk össze!
- **Tk.2.4.14. (54-57.o.) 2.4.1.feladata Termékek feladatai**  
[http://people.inf.elte.hu/sila/AB1ea/Feladatok\\_Termekek.pdf](http://people.inf.elte.hu/sila/AB1ea/Feladatok_Termekek.pdf)  
create table: [http://people.inf.elte.hu/sila/eduAB/create\\_termekek.txt](http://people.inf.elte.hu/sila/eduAB/create_termekek.txt)
- **Tk.2.4.14. (57-60.o.) 2.4.3.feladata Csatahajók feladatai**  
[http://people.inf.elte.hu/sila/AB1ea/Feladatok\\_Csatahajok.pdf](http://people.inf.elte.hu/sila/AB1ea/Feladatok_Csatahajok.pdf)  
create table: [http://people.inf.elte.hu/sila/eduAB/create\\_csatahajok.txt](http://people.inf.elte.hu/sila/eduAB/create_csatahajok.txt)