

5.előadás: Adatbázisok-I.

dr. Hajas Csilla (ELTE IK)
<http://sila.hajas.elte.hu/>

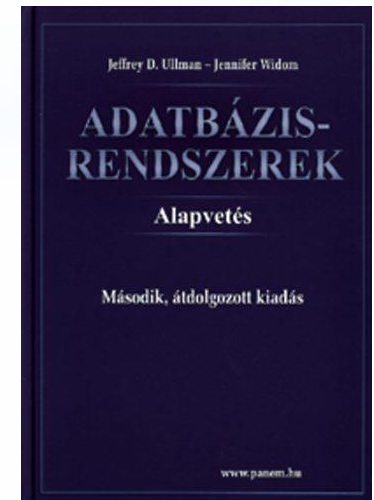
Relációs algebra és az SQL SELECT

Tankönyv:

2.4. Relációs algebra, lekérdező nyelv

6.2. Több táblás lekérdezések SQL-ben

+ Példák: „Termékek” 2.4. végén a feladatok megoldása algebrai nyelven és SQL-ben



Relációs algebrai lekérdező nyelv ---1

- **Nyelv:** a kérdés szintaktikai alakja és a kérdés kiértékelése (algoritmus) kiértékelési szemantika
- **Relációs algebra** egyszerű és hatékony módszer adott relációkból új relációk létrehozására.
- Algebra **műveleteket** és **atomi operandusokat** tartalmaz, például számtani algebra esetén műveletek (összeadás, kivonás, szorzás, osztás) és operandusok (pl. x változó, és 15-höz hasonló konstansok), kifejezések megadása.
- **Relációs algebra:** az atomi operandusokon és az algebrai kifejezéseken végzett műveletek alkalmazásával kapott relációkon műveleteket adunk meg, kifejezéseket építünk (a kifejezés felel meg a kérdés szintaktikai alakjának). Fontos tehát, hogy **minden művelet eredménye reláció**, amelyen további műveletek adhatók meg.

Relációs algebrai lekérdező nyelv ---2

Relációs algebrai kifejezés, mint L lekérdező nyelv

- Adott az adatbázis sémája: $\mathbb{R} = \{R_1, \dots, R_k\}$
 $q \in L \quad q: R_1, \dots, R_k \rightarrow V$ (eredmény-reláció)
E - relációs algebrai kifejezés: $E(R_1, \dots, R_k) = V$ (output)

Relációs algebrai kifejezések formális felépítése

- Atomi operandusai a következők:
 - a relációkhoz tartozó **változók**, $R_i \in \mathbb{R}$ (az adatbázis-sémában levő relációnevek, ahol az R_i kiértékelése: az aktuális előfordulása)
 - **konstansok**, amelyek véges relációt fejeznek ki.
- Műveletei:
 - (egy relációra) **unér**: vetítés, kiválasztás, [átnevezés];
 - (két rel.) **binér**: halmazműveletek, összekapcsolások.

Relációs algebrai lekérdező nyelv ---3

Relációs algebrai kifejezések felépítése

- Elemi kifejezések (atomi operandusok) kifejezések
- Összetett kifejezések az alábbiak:
- 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

Vetítés (project, jelölése Π)

- **Vetítés (projekció).** Adott relációt vetít le az alsó indexben szereplő attribútumokra (attribútumok számát csökkentik)
- $\Pi_{\text{lista}}(R)$ ahol lista: $\{A_{i_1}, \dots, A_{i_k}\}$ R-sémájában levő attribútumok egy részhalmazának felsorolása
eredmény típusa $\langle A_{i_1} : \text{értéktípus}_{i_1}, \dots, A_{i_k} : \text{értéktípus}_{i_k} \rangle$
 $\Pi_{\text{lista}}(R) := \{ t.A_{i_1}, t.A_{i_2}, \dots, t.A_{i_k} \mid t \in R \} = \{ t[\text{lista}] \mid t \in R \}$
- Reláció soraiból kiválasztja az attribútumoknak megfelelő A_{i_1}, \dots, A_{i_k} -n előforduló értékeket, ha többször előfordul akkor a duplikátumokat kiszűrjük (hogy halmazt kapjunk)

➤ **Példa:**

A	B	C
a	b	c
c	d	e
c	d	d

$\Pi_{A, B}(R)$



A	B
a	b
c	d

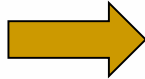
Kiválasztás (select, jelölése: σ)

- **Kiválasztás (szűrés).** Kiválasztja az argumentumban szereplő reláció azon sorait, amelyek eleget tesznek az alsó indexben szereplő feltételnek.
- $\sigma_{\text{Feltétel}}(R)$ és R sémája megegyezik
- $\sigma_{\text{Feltétel}}(R) := \{ t \mid t \in R \text{ és } t \text{ kielégíti } \sigma \text{-ban szereplő Feltételt} \}$
- $R(A_1, \dots, A_n)$ séma feletti reláció esetén a σ_F kiválasztás F feltétele a következőképpen épül fel:
 - **elemi feltétel:** $A_i \theta A_j$, $A_i \theta c$, ahol c konstans, θ pedig $=, \neq, <, >, \leq, \geq$
 - **összetett feltétel:** ha B_1, B_2 feltételek, akkor $\neg B_1, B_1 \wedge B_2, B_1 \vee B_2$ és zárójelezésekkel is feltételek

➤ **Példa:**

A	B	C
a	b	c
c	d	e
g	a	d

$\sigma_{A='a' \vee C \leq 'd'}(R)$



A	B	C
a	b	c
g	a	d

Halmazműveletek (jelölése a szokásos)

- Reláció előfordulás **véges sok sorból álló halmaz**. Így értelmezhetők **a hagyományos halmazműveletek**: **unió** (az eredmény halmaz, csak egyszer szerepel egy sor) **halmazműveleti különbség**, **metszet**. Milyen művelet van még halmazokon? Vajon ez értelmezhető-e relációkon?
- R, S és azonos típusú, $R \cup S$ és $R - S$ típusa ugyanez
 $R \cup S := \{t \mid t \in R \vee t \in S\}$, $R - S := \{t \mid t \in R \wedge t \notin S\}$
- Az alpműveletekhez az **unió** és **különbség** tartozik, **metszet** műveletet származtatjuk $R \cap S = R - (R - S)$

A	B	C
a	b	c
c	d	e
g	a	d

A	B	C
a	b	c
c	d	e
g	d	f

Példa: különbségre

$R - S$



A	B	C
g	a	d

Példák: Halmazműveletek

Felhasználó1:

Söröző	Sör	Ár
Joe's	Bud	2.50
Joe's	Miller	2.75
Sue's	Bud	2.50

Felhasználó2:

Söröző	Sör	Ár
Joe's	Bud	2.50
Jack's	Bud	2.75

Felhasználó1 \cup Felhasználó2:

Söröző	Sör	Ár
Joe's	Bud	2.50
Joe's	Miller	2.75
Sue's	Bud	2.50
Jack's	Bud	2.75

Felhasználó1 \cap Felhasználó2:

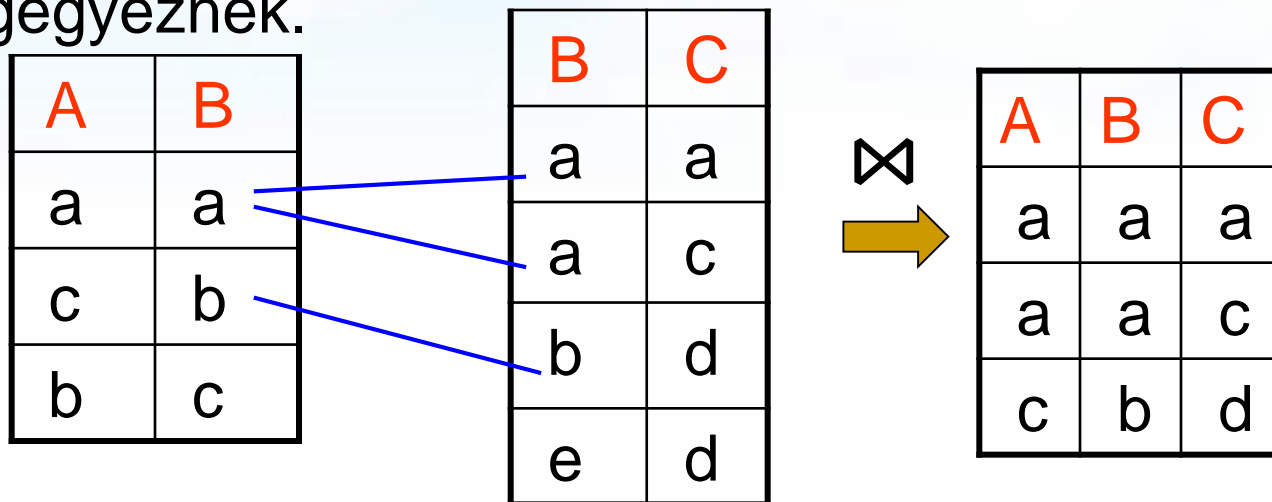
Söröző	Sör	Ár
Joe's	Bud	2.50

Felhasználó2 \setminus Felhasználó1:

Söröző	Sör	Ár
Jack's	Bud	2.75

Természetes összekapcsolás ---1

- Szorzás jellegű műveletek (attribútumok számát növeli) a relációs algebrai lekérdezésekben, kifejezésekben többféle lehetőséget is használhatunk, ebből az egyik alpművelet:
- **Natural join** (angol neve, jelölése: \bowtie „csokornyakkendő”)
- **Természetes összekapcsolás** (magyar elnevezése) közös attribútum-nevekre épül. $R \bowtie S$ azon sorpárokat tartalmazza R-ből illetve S-ből, amelyek R és S azonos attribútumain megegyeznek.



Természetes összekapcsolás ---2

➤ Természetes összekapcsolás:

Legyen $R(A_1, \dots, A_k, B_1, \dots, B_n)$, illetve $S(B_1, \dots, B_n, C_1, \dots, C_m)$

➤ $R \bowtie S$ típusa $(A_1, \dots, A_k, B_1, \dots, B_n, C_1, \dots, C_m)$ vagyis a két attribútum-halmaz uniója

➤ $R \bowtie S$ elemei, vagyis sorai $v \in R \bowtie S$

$$R \bowtie S = \{ v \mid \exists t \in R, \exists s \in S: t[B_1, \dots, B_n] = s[B_1, \dots, B_n] \wedge v[A_1, \dots, A_k] = t[A_1, \dots, A_k] \wedge v[B_1, \dots, B_n] = t[B_1, \dots, B_n] \wedge v[C_1, \dots, C_m] = s[C_1, \dots, C_m] \}$$

Természetes összekapcsolás ---3

- **Példákban:** két azonos nevű attribútumot úgy tekintünk, hogy ugyanazt jelenti és a közös érték alapján fűzzük össze a sorokat.
- **Milyen problémák lehetnek?**
- Filmek adatbázisban ugyanarra a tulajdonságra más névvel hivatkozunk: Filmek.év és SzerepelBenne.filmÉv, illetve FilmSzínész.név és SzerepelBenne.színészNév
- Termékek adatbázisban pedig ugyanaz az azonosító mást jelent: Termék.típus más mint a Nyomtató.típus
- Emiatt a Filmek és a Termékek adatbázisokban ahhoz, hogy jól működjön az összekapcsolás **szükségünk van** egy technikai műveletre, az attribútumok illetve relációk **átnevezésére**

Átnevezés (rename, jelölése ró: ρ)

- Miért van erre szükség? Nem tudjuk a reláció saját magával való szorzatát kifejezni, $R \bowtie R = R$ lesz.
- Láttuk, hogy egyes esetekben szükség lehet relációnak vagy a reláció attribútumainak **átnevezésére**:

$$\rho_{S(B_1, \dots, B_k)}(R(A_1, \dots, A_k))$$

- Ha csak a relációt nevezzük át, ezt $\rho_S(R)$ -rel jelöljük. Ha ugyanazt a táblát használjuk többször, akkor a táblának adunk másik hivatkozási (alias) nevet.
- Az attribútumok átnevezése helyett alternatíva: **R.A** (vagyis relációnév.attribútumnév hivatkozás) amivel meg tudjuk különböztetni a különböző táblákból származó azonos nevű attribútumokat.

Szorzás jellegű műveletek ---1

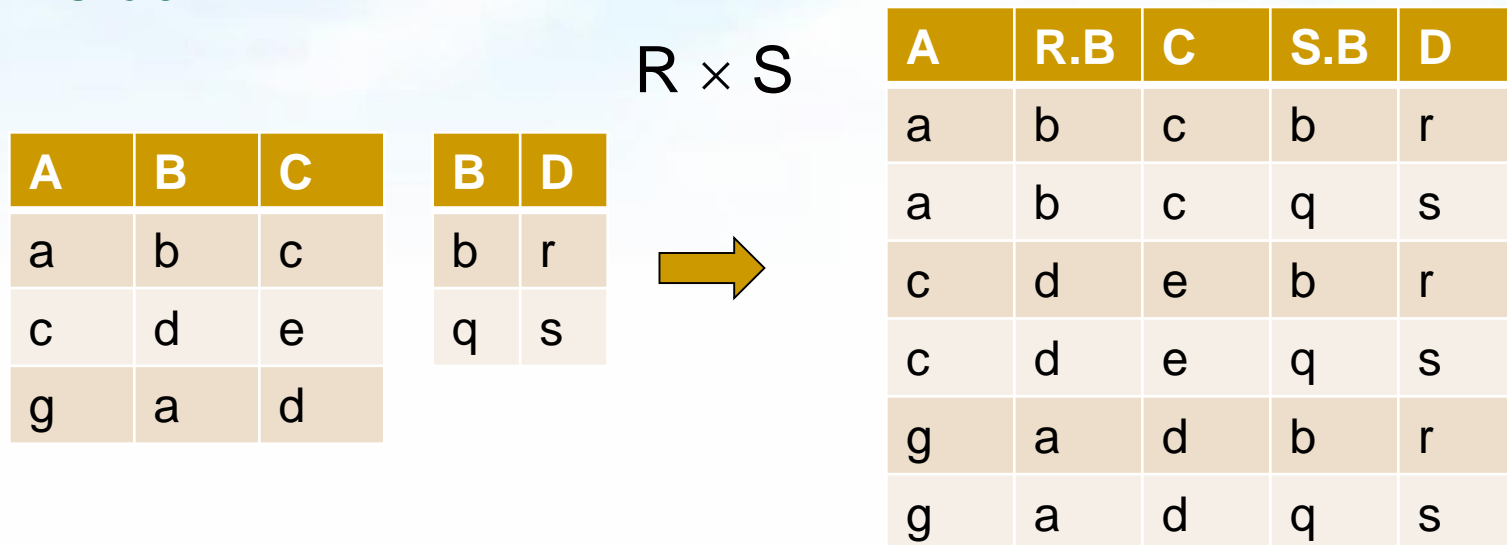
- További szorzás jellegű műveletek, amelyek két reláció sorait kombinálják: **Descartes-szorzat (direkt-szorzat)**
Ezt is használjuk a relációs algebrai lekérdezésekben és lekérdezési tervekben, de a sokkal gyakrabban használt **természetes összekapcsolást** tekintjük alpműveletnek.
- **Descartes-szorzat $R \times S$** : az R és S minden sora párban összefűződik, az első tábla minden sorához hozzáfűzzük a második tábla minden sorát.
Legyen $R(A_1, \dots, A_k, B_1, \dots, B_n)$, illetve $S(B_1, \dots, B_n, C_1, \dots, C_m)$
- $R \times S$ típusa $(A_1, \dots, A_k, R.B_1, \dots, R.B_n, S.B_1, \dots, S.B_n, C_1, \dots, C_m)$
vagyis a két attribútum-halmaz multihalmazként való uniója
- $R \times S$ elemei, sorai: $R \times S := \{ t \mid t[R] \in R \text{ és } t[S] \in S \}$

Szorzás jellegű műveletek ---2

- **Descartes-szorzat** (vagy szorzat, **direkt-szorzat**) esetén természetesen nem fontos az attribútumok egyenlősége. A két vagy több reláció azonos nevű attribútumait azonban meg kell különböztetni egymástól.

Hivatkozás séma: oszlopok átnevezése illetve azonos nevű oszlop esetén: $R.B_1, \dots, R.B_n, S.B_1, \dots, S.B_n$

Példa:



Szorzás jellegű műveletek ---3

Felhasználó:

Söröző	Sör	Ár
Joe's	Bud	2.50
Joe's	Miller	2.75
Sue's	Bud	2.50
Sue's	Miller	3.00

Théta-join \bowtie_{θ}

Söröző:

Név	Cím
Joe's	Maple st.
Sue's	River rd.

Barinfo = Felhasználó $\bowtie_{\text{Felhasználó.söröző} = \text{Söröző.név}}$ Söröző

Söröző	Sör	Ár	Név	Cím
Joe's	Bud	2.50	Joe's	Maple st.
Joe's	Miller	2.75	Joe's	Maple st.
Sue's	Bud	2.50	Sue's	River rd.
Sue's	Miller	3.00	Sue's	River rd.

Szorzás jellegű műveletek ---4

- Ha R, S sémái megegyeznek, akkor $R \bowtie S = R \cap S$.
- Ha R, S sémáiban **nincs közös attribútum**, akkor $R \bowtie S = R \times S$.
- **Feladatok:** Hogyan fejezhető ki az $R \times S$ **direkt szorzat** relációs algebrában? (a **természetes összekapcsolást** tekintjük alpműveletnek, ebből és az átnevezés segítségével felírható a direkt szorzat).
- Hogyan fejezhető ki **a természetes összekapcsolás**, ha a **direkt szorzatot** sorolnánk az alpműveletek közé?
- **Köv.előadáson folytatjuk:** Théta összekapcsolás \bowtie_{θ} , stb
Alkérdések segítségével félig összekapcsolások \bowtie is,
Rel.algebra kiterjesztése **külső összekapcsolások** is!

6.2. Több táblára vonatkozó lekérdezések az SQL-ben

Select-From-Where (SFW) utasítás

- Gyakran előforduló relációs algebrai kifejezés
 $\Pi_{\text{Lista}} (\sigma_{\text{Felt}} (R_1 \times \dots \times R_n))$ típusú kifejezések
 - **Szorzat és összekapcsolás az SQL-ben**
 - **SELECT** s-lista -- milyen típusú sort szeretnénk az eredményben látni?
FROM f-lista -- relációk (táblák) összekapcsolása, illetve szorzata
WHERE felt -- milyen feltételeknek eleget tevő sorokat kell kiválasztani?
 - **FROM f-lista** elemei (ezek ismétlődhetnek)
táblanév [[AS] sorváltozó, ...]
- Itt: a from lista elemei a táblák direkt szorzatát jelenti, az összekapcsolási feltételt where-ben adjuk meg, később bevezetünk majd további lehetőségeket a különböző összekapcsolásokra az SQL from záradékában.

Attribútumok megkülönböztetése ---1

- **Milyen problémák merülnek fel?**
- (1) Ha egy attribútumnév több sémában is előfordul, akkor nem elég az attribútumnév használata, mert ekkor nem tudjuk, hogy melyik sémához tartozik.
- Ezt a problémát az SQL úgy oldja meg, hogy megengedi egy relációnévnek és egy pontnak a használatát egy attribútum előtt: **R.A** (az R reláció A attribútumát jelenti).
- **Természetes összekapcsolás** legyen R(A, B), S(B,C)

```
SELECT A, R.B B, C
FROM R, S
WHERE R.B=S.B;
```

Attribútumok megkülönböztetése ---2

- Milyen problémák merülnek még fel?
- (2) Ugyanaz a reláció többször is szerepelhet, vagyis szükség lehet arra, hogy ugyanaz a relációnév többször is előforduljon a FROM listában.
- Ekkor a FROM listában a táblához másodnevet kell megadni, erre **sorváltzóként** is szoktak hivatkozni, megadjuk, h. melyik sorváltzó melyik relációt képviseli:
FROM $R_1 [t_1], \dots, R_n [t_n]$
Ekkor a SELECT és WHERE záradékok kifejezésekben a hivatkozás: **$t_i.A$** (vagyis sorváltzó.attribútumnév)

Példa: Két tábla összekapcsolása ---1

- Mely söroket szeretik a Joe's Bárba járó sörivók?

```
SELECT sör
```

```
FROM Szeret, Látogat
```

```
WHERE söröző = 'Joe' 's Bar'
```

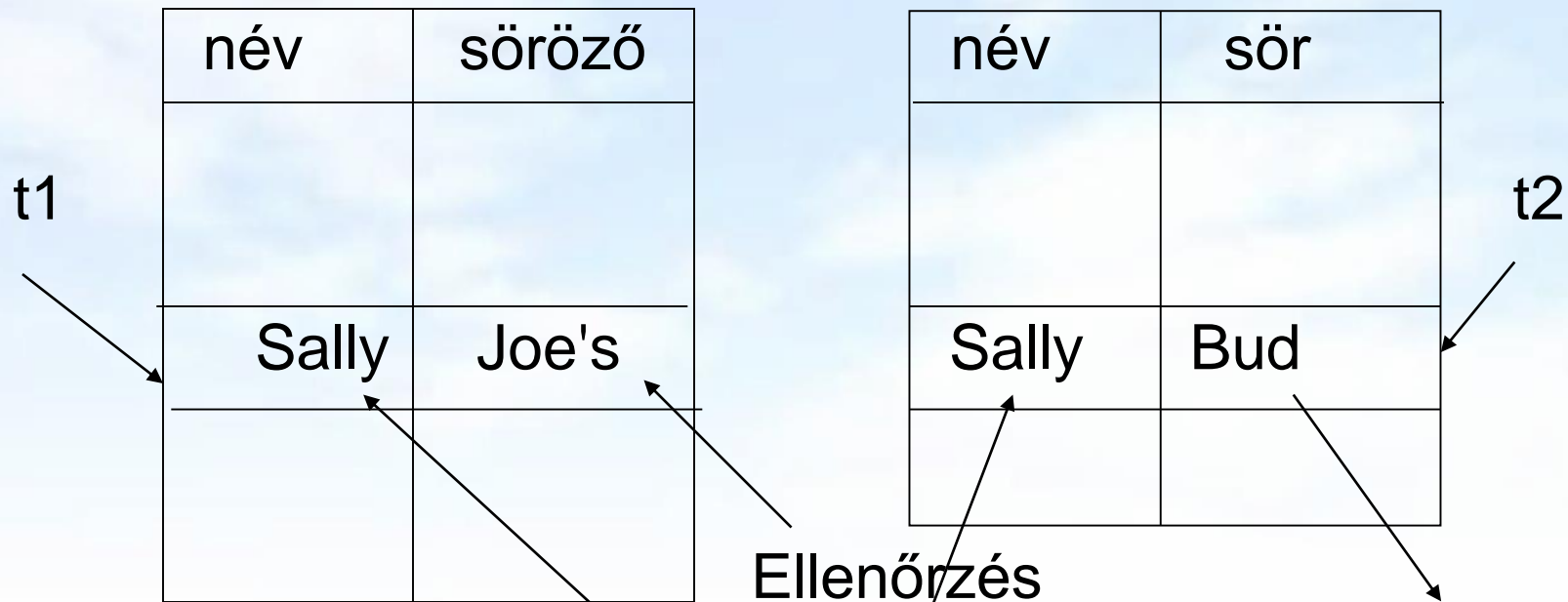
```
AND Látogat.név = Szeret.név;
```

- Kiválasztási feltétel: **söröző = 'Joe' 's Bar'**
- Összekapcsolási feltétel: **Látogat.név = Szeret.név**
- Alapértelmezése a következő oldalon a mai órán
- Összekapcsolások új szintaxisát köv.órán nézzük majd

Példa: Két tábla összekapcsolása ---2

Látogat

Szeret



Ellenőrzés
Joe's bárja

Ellenőrizzük, hogy
megegyeznek-e

SFW szabvány alapértelmezése ---1

- Kiindulunk a **FROM záradékból**: a FROM lista minden eleméhez **egy beágyazott ciklus** végigfut az adott tábla sorain a ciklus minden lépésénél az n darab sorváltozónak lesz egy-egy értéke
- ehhez kiértékeljük a WHERE feltételt, vagyis elvégezzük a **WHERE záradékban** szereplő feltételnek eleget tevő sorok kiválasztását (csak a helyesek, ahol TRUE=igaz választ kapunk), azok a sorok kerülnek az eredménybe.
- Alkalmazzuk a **SELECT záradékban** jelölt kiterjesztett projekciót. Az **SQL-ben az eredmény alapértelmezés szerint** itt sem halmaz, hanem **multihalmaz**.

Ahhoz, hogy halmazt kapjunk, azt külön kérni kell:
SELECT DISTINCT Lista

SFW szabvány alapértelmezése ---2

FOR t_1 sorra az R_1 relációban DO

FOR t_2 sorra az R_2 relációban DO

...

FOR t_n sorra az R_n relációban DO

IF a where záradék igaz, amikor az attribútumokban
 t_1, t_2, \dots, t_n megfelelő értékei találhatóak

THEN

t_1, t_2, \dots, t_n -nek megfelelően kiértékeljük a
select záradék attribútumait
és az értékekből alkotott sort
az eredményhez adjuk

SFW szabvány alapértelmezése ---3

```
SELECT [DISTINCT] kif1 [[AS] onév1], ..., kifn [[AS] onévn]  
FROM R1 [t1], ..., Rn [tn]  
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
- Ha helyes (vagyis igaz) választ kaptunk, akkor képezünk egy sort a SELECT záradékban szereplő kifejezéseknek megfelelően.

Megj.: konverzió relációs algebrába

SELECT [DISTINCT] kif₁ [[AS] onév₁], ..., kif_n [[AS] onév_n]
FROM R₁ [t₁], ..., R_n [t_n]

WHERE feltétel (vagyis logikai kifejezés)

- 1.) A FROM záradék sorváltozóiból indulunk ki, és tekintjük a hozzájuk tartozó relációk Descartes-szorzatát. Átnevezéssel valamint R.A jelöléssel elérjük, hogy minden attribútumnak egyedi neve legyen.
- 2.) A WHERE záradékot átalakítjuk egy kiválasztási feltétellé, melyet alkalmazunk az elkészített szorzatra.
- 3.) Végül a SELECT záradék alapján létrehozzuk a kifejezések listáját, a (kiterjesztett) vetítési művelethez.

$$\Pi_{\text{onév}_1, \dots, \text{onév}_n} (\sigma_{\text{feltétel}} (R_1 \times \dots \times R_n))$$

Tábla önmagával való szorzata ---1

- Bizonyos lekérdezéseknél arra van szükségünk, hogy ugyanannak a relációnak több példányát vegyük.
- Ahhoz, hogy meg tudjuk különböztetni a példányokat a relációkat átnevezzük, másodnevet adunk, vagyis **sorváltozókat** írunk mellé a FROM záradékban.
- A relációkat mindig átnevezhetjük ily módon, akkor is, ha egyébként nincs rá szükség (csak kényelmesebb).
- **Példa: R(Szülő, Gyerek)** séma feletti relációban adott szülő-gyerek adatképekből állítsuk elő a megállapítható Nagyszülő-Unoka párokat!

```
SELECT t1.Szülő NagySzülő, t2.Gyerek Unoka  
FROM R t1, R t2  
WHERE t1.Gyerek = t2.Szülő;
```

Tábla önmagával való szorzata ---2

- **Példa: Sörök(név, gyártó)** tábla felhasználásával keressük meg az összes olyan sörpárt, amelyeknek ugyanaz a gyártója.
 - Ne állítsunk elő (Bud, Bud) sörpárokat.
 - A sörpárokat ábécé sorrendben képezzük, például ha (Bud, Miller) szerepel az eredményben, akkor (Miller, Bud) ne szerepeljen.

```
SELECT s1.név, s2.név  
FROM Sörök s1, Sörök s2  
WHERE s1.gyártó = s2.gyártó  
AND s1.név < s2.név;
```

Halmazműveletek az SQL-ben

- Mi hiányzik még, hogy a relációs algebra alpműveleteit mindet az SQL-ben vissza tudjuk adni?
- A relációs algebrai halmazműveletek: **unió, különbség** mellett az **SQL-ben ide soroljuk a metszetet is** (ugyanis fontos a metszet és az SQL-ben is implementálva van).
- Az SQL-ben a halmazműveleteket úgy vezették be, hogy azt mindig két lekérdezés között lehet értelmezni, vagyis nem relációk között, mint $R \cup S$, hanem lekérdezem az egyiket is és a másikat is, majd a lekérdezések unióját veszem.

(SFW-lekérdezés1)

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

(SFW-lekérdezés2);

- **Köv.előadáson** lesz a multihalmazokra való kiterjesztése!

Példa: Intersect (metszet)

- Szeret(név, sör), Felszolgál(söröző, sör, ár) és Látogat(név, söröző) táblák felhasználásával keressük

Trükk: itt ez az az alkérdés valójában az adatbázisban tárolt tábla azokat (név,sör) párokat, ahol a név = sörivó látogat olyan sörözőt, ahol felszolgálnak olyan sört, amelyet szeret (a „boldog” sörivók).

(SELECT * FROM Szeret)

(név, sör) párok, ahol a sörivó látogat olyan sörözőt, ahol ezt a sört felszolgálják

INTERSECT

(SELECT név, sör

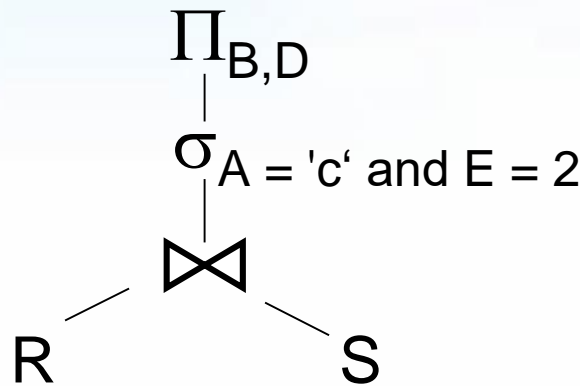
FROM Látogat L, Felszolgál F

WHERE L.söröző = F.söröző);

Példák, gyakorló feladatok több táblára vonatkozó lekérdezésekre relációs algebrában és SQL-ben

Lekérdezések kifejezése algebrában ---1

- **A relációs algebra** procedurális nyelv, vagyis nemcsak azt adjuk meg, hogy **mit** csináljunk, hanem azt is **hogyan**.
- Kifejezés kiértékelése: összetett kifejezést kívülről befelé haladva átírjuk kiértékelő fává, levelek: elemi kifejezések.
- 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)



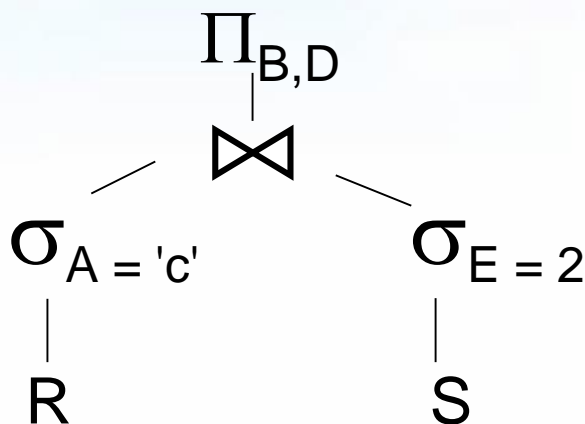
- Tudunk-e ennél jobb, hatékonyabb megoldást találni?

Lekérdezések kifejezése algebrában ---2

- **Ekvivalens átalakítási lehetőségekkel**, relációs algebrai azonosságokkal (például mikor cserélhető fel a \bowtie természetes összekapcsolás és a σ_{felt} kiválasztás?) át tudjuk alakítani a fentivel ekvivalens másik relációs algebrai kifejezésre. Kérdés: 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.
- Először **táblákkal gondolkodva** nézzük meg, hogy milyen stratégiákkal, milyen tábla műveletekkel tudjuk megkapni az output táblát, a kívánt eredményt, ezt írjuk fel relációs algebrában lineáris módon és kifejezőfával, majd SQL-ben!
- 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.

Példa: Termékek (Tk.2.4.1.feladat) --- 1

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?

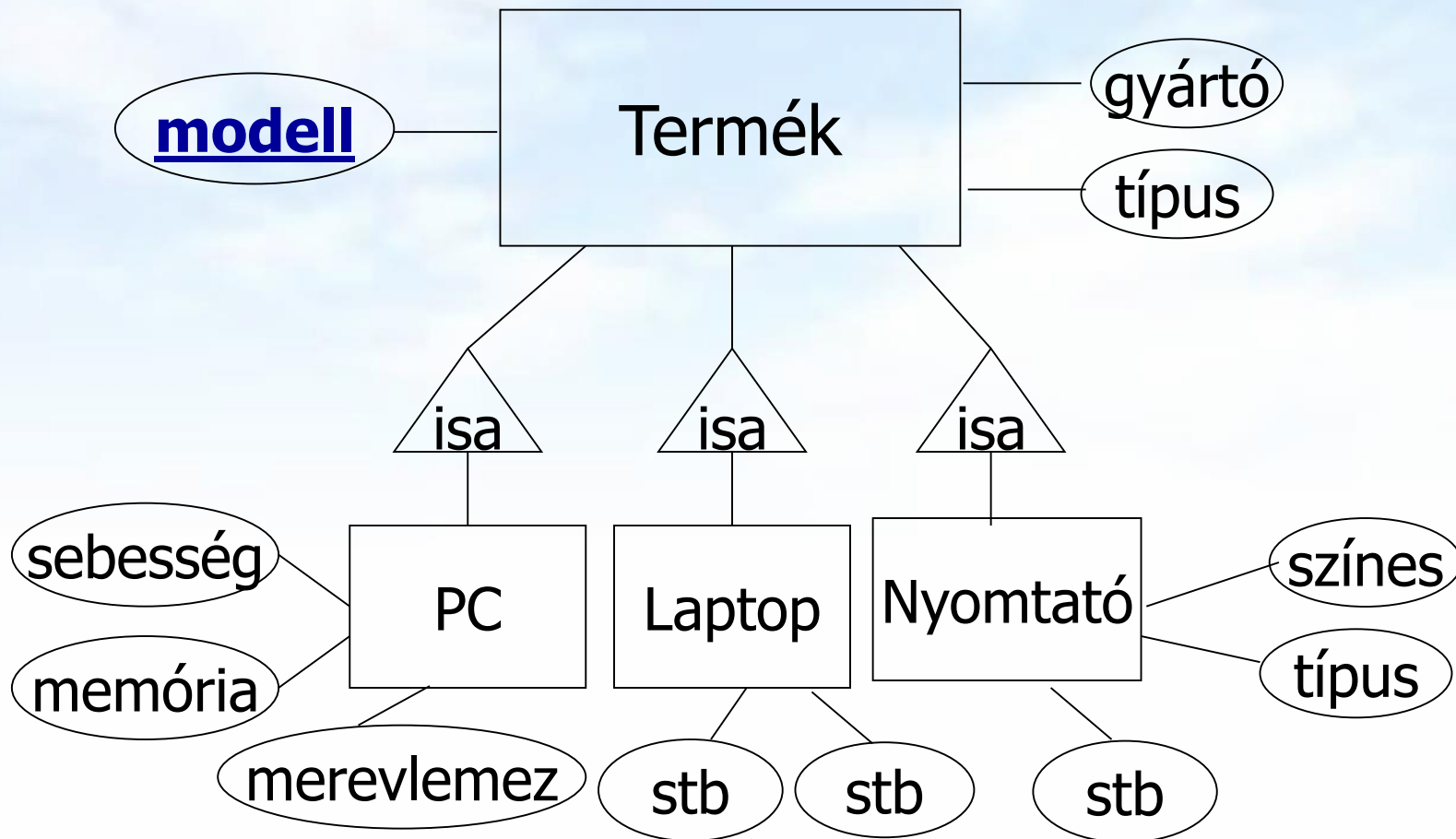
c) Adjuk meg a B gyártó által gyártott összes termék modellszámát és árát!
stb...

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

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

Példa: Termékek (Tk.2.4.1.feladat) --- 2

Feladat: Az alábbi E/K diagramot írjuk át relációsémákra
Nézzünk meg több megközelítéssel (1 v. 3 v. 4 táblára is)



Példa: Termékek (Tk.2.4.1.feladat) --- 3

- 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(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”

Probléma: természetes összekapcsolás

Termek táblának modell elsődleges kulcsára hivatkozik a Nyomtato táblában modell a külső kulcs (hivatkozás)

Természetes összekapcsolás: Itt hibás eredményt kapunk!

```
SELECT modell, gyarto, tipus
```

```
FROM Termek NATURAL JOIN Nyomtato;
```

-- Hiba: modell, tipus (két oszlopnak is megegyezik a neve)

-- mivel a tipus mást jelent a két táblában, ezért ÜRES lesz

Köv.héten megnézzük a natural join művelet új szintaxisát, ahol megadhatjuk a kapcsolómezőket, ez jó megoldást ad:

```
SELECT modell, gyarto, T.tipus, N.tipus
```

```
FROM Termek T JOIN Nyomtato N
```

```
USING ( modell );
```

Előző probléma megoldása

Descartes szorzattal (FROM listán a táblákat megadjuk, és a WHERE-be írjuk be a kapcsolást, itt a DBMS joint végez)

```
SELECT T.modell, gyarto, T.tipus, N.tipus  
FROM Termek T, Nyomtato N  
WHERE T.modell = N.modell;
```

Köv.héten ebben a szemléletben is lesz az új szintaxissal:

```
SELECT T.modell, gyarto, T.tipus, N.tipus  
FROM Termek T JOIN Nyomtato N  
ON T.modell = N.modell;
```

Köv.előadáson az összekapcsolás további lehetőségeit is átvesszük, például alkérdésekkel (algebrában: szemijoin)

Példák relációs algebrai lekérdezésekre ---a.)

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

Példák átírásokra ---a.)

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

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

SELECT modell

FROM PC

WHERE sebesseg >= 3.00;

Π_m

|

$\sigma_{s \geq 3.00}$

|

PC

Példák relációs algebrai lekérdezésekre ---b.)

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

Példák relációs algebrai lekérdezésekre ---b.)

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

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

Példák átírásokra ---b.)

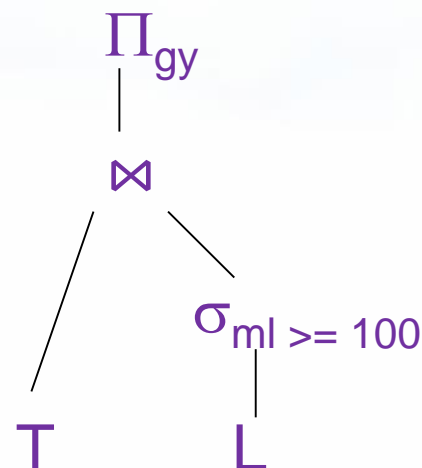
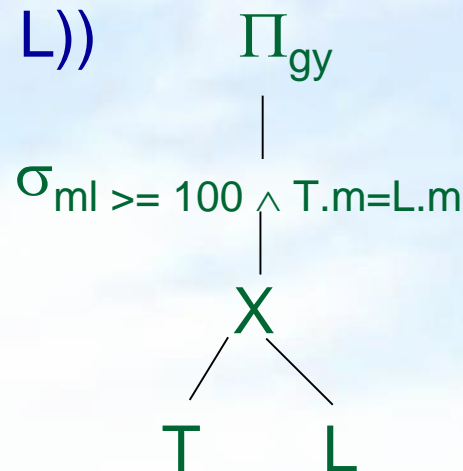
b.) Mely gyártók készítenek legalább száz gigabájt méretű merevlemezzel rendelkező laptopot?

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

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;

-- köv.előadáson (alkérdésekkel)
SELECT gyarto FROM Termek
WHERE modell IN
(SELECT modell FROM Laptop
WHERE merevlemez >= 100)



Példák relációs algebrai lekérdezésekre ---c.)

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

Példák relációs algebrai lekérdezésekre ---c.)

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!)

-- keressünk több megoldást, különböző nézőpontokból!

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

$$\prod_{m, ár}(\mathbf{BT} \bowtie \mathbf{PC}) \cup$$

$$\prod_{m, ár}(\mathbf{BT} \bowtie \mathbf{L}) \cup$$

$$\prod_{m, ár}(\mathbf{BT} \bowtie \mathbf{Ny})$$

Példák átírásokra ---c.)

c.) Adjuk meg a B gyártó által gyártott összes termék modellszámát és árát típustól függetlenül! 1.mego.:

BT := $\Pi_m \sigma_{gy='B'}(T)$ munkatáblával $\Pi_{m, ár}(\mathbf{BT} \bowtie \mathbf{PC}) \cup$
 $\Pi_{m, ár}(\mathbf{BT} \bowtie \mathbf{L}) \cup$
 $\Pi_{m, ár}(\mathbf{BT} \bowtie \mathbf{Ny})$

with BT as

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

select modell, ar from pc natural join **BT**

union

select modell, ar from laptop natural join **BT**

union

select modell, ar from nyomtato natural join **BT;**

Példák átírásokra ---c.)

c.) Adjuk meg a B gyártó által gyártott összes termék modellszámát és árát típustól függetlenül! 2.mego.:

-- más szemlélettel más segédváltozót vezetek be

Arlista := $\Pi_{m, \text{ár}}(\mathbf{PC}) \cup \Pi_{m, \text{ár}}(\mathbf{L}) \cup \Pi_{m, \text{ár}}(\mathbf{Ny})$

with **Arlista** as

(select modell, ar from pc

union

select modell, ar from laptop

union

select modell, ar from nyomtato)

select modell, ar from **Arlista** natural join Termek
where gyarto='B';

Példák relációs algebrai lekérdezésekre ---d.)

d.) Adjuk meg valamennyi színes lézernyomtató modellszámát

Példák relációs algebrai lekérdezésekre ---d.)

d.) Adjuk meg valamennyi színes lézernyomtató

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

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

-- és elvégezhető a tábla önmagával való szorzatával is:

$= \Pi_{Ny1.m}(\sigma_{Ny1.sz='i' \wedge Ny2.t='lézer' \wedge Ny1.m=Ny2.m}(Ny1 \times Ny2))$

!!! Hasonlítsuk össze azzal, amikor szeret(Név, Gyümölcs) sémájú reláció alapján keressük, hogy kik szeretik az almát is és a körtét is! A fenti relációs algebrai megoldások közül melyik ad helyes/illetve téves eredményt erre a kérdésre?

Példák átírásokra ---d.)

d.) Adjuk meg valamennyi színes lézernyomtató

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

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

select modell from Nyomtato --- 1.megoldás (metszet)
where tipus='lézer'

intersect

select modell from Nyomtato
where szines='igen';

select modell from Nyomtato --- 2.mego.: ekvivalens-e?
where tipus='lézer' **and** szines='igen';

--- Hogyan tudjuk átírni SQL-be a többit (többféleképpen)

Példák relációs algebrai lekérdezésekre ---e.)

e) Melyek azok a gyártók, amelyek laptopot árulnak,
PC-t viszont nem?

Példák relációs algebrai lekérdezésekre ---e.)

- e) Melyek azok a gyártók, amelyek laptopot árulnak, PC-t viszont nem? (itt elég lenne csak a Termék táblát használni, mert abban benne van a termék típusa: Ez, akkor az a feladat lenne, mint a Szeret(név, gyümölcs) tábla alapján adjuk meg „kik azok, akik szeretik az almát, de nem szeretik a körtét” típusú lekérdezési feladat).
- Vizsgáljuk meg a kérdést! Ha a Termék táblát bővítjük olyan PC modellekkel, akiknek a gyártója laptopot gyárt, akkor az alaptábla bővítésével, vagyis új sorok felvitelével az eredménytábla csökken. Az ilyen típusú lekérdezések **nem monoton** jellegűek, ezek az „Elhagyásos-feladatok”, az alap relációs algebraiban az egyetlen **nem monoton** művelet a halmazműveleti különbség szükséges: **R – S**

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

Példák relációs algebrai lekérdezésekre ---f.)

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

Példák relációs algebrai lekérdezésekre ---f.)

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

--- amikor egy táblát önmagával szorozzuk, akkor átnevezzük a táblákat: **P1 := PC, P2 := PC**

$\Pi_{p1.ml}(\sigma_{p1.m \neq p2.m \wedge p1.ml = p2.ml} (P1 \times P2))$

Példák átírásokra ---f.)

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

--- átnevezzük a táblákat **P1 := PC, P2 := PC**

$\Pi_{p1.ml}(\sigma_{p1.m \neq p2.m \wedge p1.ml = p2.ml} (P1 \times P2))$

```
select distinct p1.merevlemez
from   PC p1, PC p2
where  p1.modell != p2.modell
and    p1.merevlemez = p2.merevlemez;
```


Példák relációs algebrai lekérdezésekre ---g.)

! 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 ---g.)

! 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.

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

--- átírása SQL-be hasonló, mint f.)

Példák relációs algebrai lekérdezésekre ---h.)

!! 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 ---h.)

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

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

-- és ezzel legyen: **T₁** := $T \bowtie \text{Gyors}$ és **T₂** := $T \bowtie \text{Gyors}$

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

Példák relációs algebrai lekérdezésekre ---i.)

!! i) Melyik modell a leggyorsabb PC-t?

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

Példák relációs algebrai lekérdezésekre ---i.)

!! i) Melyik modell 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 ---i.)

!! i) Melyik modell a leggyorsabb PC-t?

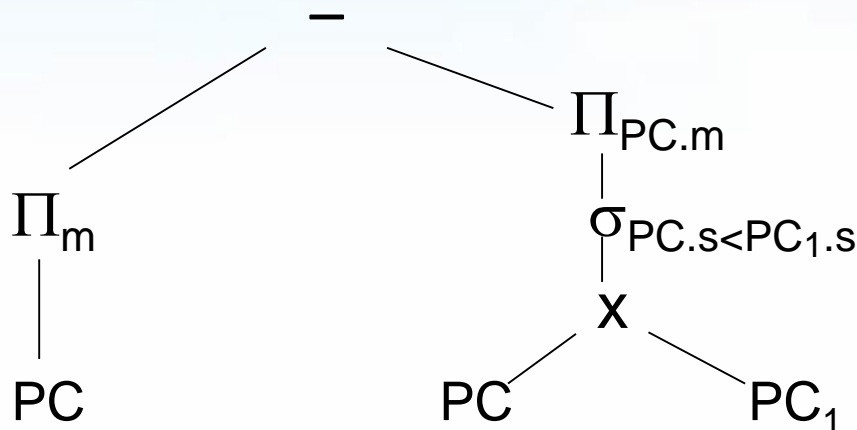
(az „elhagyás” típusú lekérdezések, lásd maximum kif.)

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 átírásokra ---i.)

!! i) Melyik modell a leggyorsabb PC?

EnnélVanNagyobb = $\Pi_{PC.m}(\sigma_{PC.s < PC_1.s}(PC \times PC_1))$
Leggyorsabb: $\Pi_m(PC) - \text{EnnélVanNagyobb}$

select modell from PC

minus

select p1.modell from PC p1, PC p2

where p1.sebesseg < p2.sebesseg;

-- Eredeti feladatban: Melyik gyártó gyártja a leggyorsabb számítógépet? Lásd köv.lapon:

Példák átírásokra ---i.)

!! i) Melyik gyártó gyártja a leggyorsabb számítógépet
Segédváltozót vezetünk be, szgep (PC vagy laptop)

with szgep as

```
( select modell, sebesseg from pc  
  union  
  select modell, sebesseg from laptop )
```

```
select gyarto, modell, sebesseg  
from termek natural join szgep  
where sebesseg =
```

```
(select sebesseg from szgep  
  minus
```

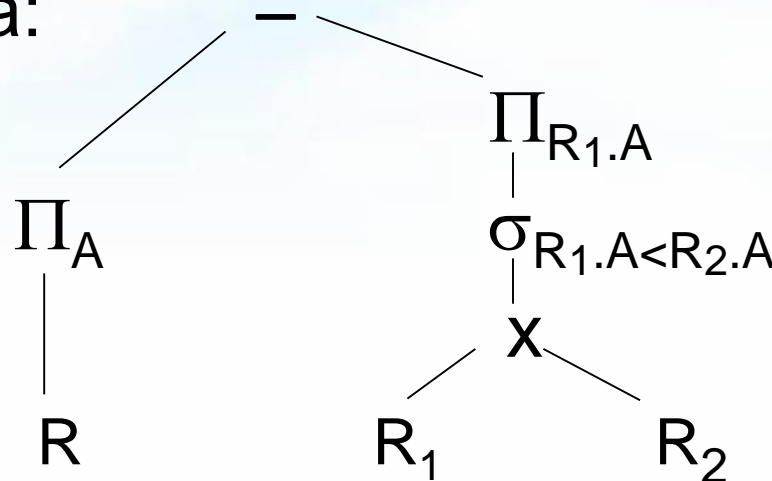
```
select p1.sebesseg from szgep p1, szgep p2  
  where p1.sebesseg < p2.sebesseg);
```

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).

- $\pi_A(R) - \pi_{R1.A}(\sigma_{R1.A < R2.A}(\rho_{R1}(R) \times \rho_{R2}(R)))$

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

- Köv.héten lesz másik megoldása, alkérdéssel:

```
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 forgalmazznak? **legalább 3-ból - legalább 4-t kivonni**

➤ Mire érdemes felhívni a figyelmet?

Mi a leggyakrabban előforduló típus, amiből építkezek?

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

Ezt a komponenst támogatja legerősebben majd az SQL:

SELECT s-lista FROM f-lista WHERE feltétel;

Összefoglalás

- Az alap relációs algebra (Tankönyv 2.4)
- SELECT - FROM - WHERE lekérdezések és halmazműveletek SQL-ben (Tankönyv 6.1-6.2)
- Először relációs algebrai nyelven táblákkal és műveletekkel többféle megoldási lehetőséget vizsgáltunk meg, amelyeket összevetettünk, majd megnéztünk SQL lekérdezésként is.
- **5.EA** (Tk. 54-57.o.) **2.4.1.feladata Termékek feladataihoz**
 - http://sila.hajas.elte.hu/AB1gy/create_termekek.txt
 - http://sila.hajas.elte.hu/AB1gy/Feladatok_Termekek.pdf
- **H.F.** (Tk. 57-60.o.) **2.4.3.feladata Csatahajók feladataihoz**
 - http://sila.hajas.elte.hu/AB1gy/create_csatahajok.txt
 - http://sila.hajas.elte.hu/AB1gy/Feladatok_Csatahajok.pdf

Kérdés/Válasz

- **Köszönöm a figyelmet! Kérdés/Válasz?**
- **Gépes gyakorlaton az Oracle Példatár feladatai**
3. fejezet: Többtáblás lekérdezések SQL-ben
- **Következő előadásokon:** Több táblás lekérdezések alkérdések használata az SQL SELECT utasítás FROM, WHERE és HAVING záradékaiban, továbbá összekapcsolások az SQL-ben (Tankönyv 6.3.)
- **Később:** Bonyolultabb lekérdezési feladatoknál a logikát (Datalog) is használjuk (Tk.5.3-5.4. és 10.2.)