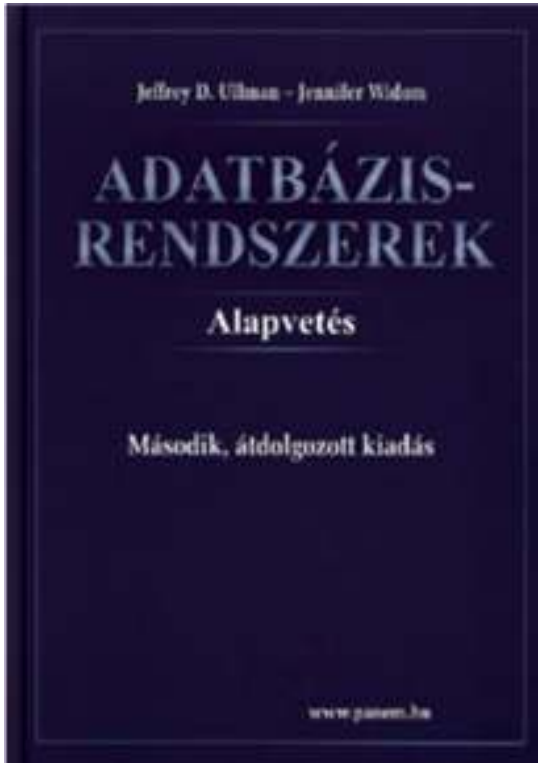
Lekérdezések az SQL-ben 5.rész

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



6.4. Relációkra vonatkozó műveletek

6.4.1. Ismétlődések megszüntetése
6.4.2. Ismétlődések kezelése
halmazműveletek során

6.4.3-6.4.7. Csoportosítás és összesítések az SQL-ben
GROUP BY és HAVING záradék

Relációkra vonatkozó műveletek

- Az eddig tanult műveleteket: vetítés (Π), kiválasztás (σ), halmazműveletek: unió ($\cup$), különbség ($-$), metszet ($\cap$), szorzás: természetes összekapcsolás($\bowtie$), direkt-szorzat (x) multihalmazok fölött értelmezzük, mint az SQL-ben, egy reláció nem sorok halmazából, hanem multihalmazából áll, vagyis megengedett a sorok ismétlődése.
- Ezeken kívül a SELECT kiegészítéseinek és záradékainak megfeleltetett új műveletekkel is kibővítjük a rel. algebrát:
  - Ismétlődések megszüntetése (δ) - select distinct ..
  - Összesítő műveletek és csoportosítás (γ_{lista}) - group by..
  - Vetítési művelet kiterjesztése (Π_{lista}) - select kif [as onev]..
  - Rendezési művelet (τ_{lista}) - order by..
  - Külső összekapcsolások ($\overset{o}{\bowtie}$) – [left | right | full] outer join

Ismétlődések megszüntetése

- SELECT DISTINCT … FROM …
- A δ művelet SQL-beli megfelelője, amellyel az eredményben kiszűrjük a duplikátumokat, vagyis multihalmazból halmazt állítunk elő.

Ismétlődések kezelése halmazművelet során

- (SELECT … FROM …)

  {UNION | INTERSECT | EXCEPT} [ALL]

  (SELECT … FROM …)

- Alapértelmezésben a halmaz-szemantika (duplikátumok szűrése)
- Az ALL kulcsszóval ezek a műveletek multihalmaz-szemantika szerint működnek.

Összesítések (aggregálás)

- SELECT listán:

  <Aggregáló művelet>(kifejezés) [[AS] onév], …

  SUM, COUNT, MIN, MAX aggregáló műveleteket, AVG (bevezették ezt is, mivel gyakran kell AVG) a SELECT záradékban alkalmazhatjuk egy oszlopra.

- COUNT(*) az eredmény sorainak számát adja meg.
- Itt is fontos a halmaz, multihalmaz megkülönböztetés.

  pl. SUM(DISTINCT R.A) csak a különböző értékűeket veszi figyelembe.
- NULL értékek használata, pl. SUM nem veszi figyelembe (implementáció függő, ellenőrizzük le a COUNT-ra - gyak.)

Példa: Összesítő függvények

- A Felszolgál(bár, sör, ár) tábla segítségével adjuk meg a Bud átlagos árát:

```
SELECT AVG(ár)
FROM Felszolgál
WHERE sör = 'Bud';
```

Ismétlődések kiküszöbölése összesítésben

- Az összesítő függvényen belül DISTINCT.
- Példa: hány *különféle* áron árulják a Bud sört?

```
SELECT COUNT(DISTINCT ár)
FROM Felszolgál
WHERE sör = 'Bud';
```

NULL értékek nem számítanak az összesítésben

- NULL nem számít a SUM, AVG, COUNT, MIN, MAX függvények kiértékelésekor.
- De ha nincs NULL értéktől különböző érték az oszlopban, akkor az összesítés eredménye NULL.
  - Kivétel: COUNT az üres halmazon 0-t ad vissza.

Példa: NULL értékek összesítésben

```
SELECT count(*)
FROM Felszolgál
WHERE sör = 'Bud';
```

A Bud sört árusító kocsmák száma.

```
SELECT count(ár)
FROM Felszolgál
WHERE sör = 'Bud';
```

A Bud sört ismert áron árusító kocsmák száma.

Csoportosítás

- SELECT …
  FROM …
  [WHERE …]
  [GROUP BY kif_1, … kif_k]
- Egy SELECT-FROM-WHERE kifejezést GROUP BY záradékkal folytathatunk, melyet attribútumok listája követ.
- A SELECT-FROM-WHERE eredménye a megadott attribútumok értékei szerint csoportosítódik, az összesítéseket ekkor minden csoportra külön alkalmazzuk.

Példa: Csoportosítás

- A Felszolgál(bár, sör, ár) tábla segítségével adjuk meg a sörök átlagos árát.

```
SELECT sör, AVG(ár)
FROM Felszolgál
GROUP BY sör;
```

sör	AVG(ár)
Bud	2.33
Miller	2.45

Példa: Csoportosítás

```
SELECT név, AVG(ár)
FROM Látogat L, Felszolgál F
WHERE sör = 'Bud'
        AND L.bár = F.bár
GROUP BY név;
```

Sörivó-
-kocsma-
-ár hármasok
a Bud sörre.

A sörivók
szerinti
csoportosítás.

A SELECT lista és az összesítések

- Ha összesítés is szerepel a lekérdezésben, a SELECT-ben felsorolt attribútumok
  1. vagy egy összesítő függvény paramétereként szerepelnek,
  2. vagy a GROUP BY attribútumlistájában is megjelennek.

Csoportok szűrése: HAVING záradék

- A GROUP BY záradékot egy HAVING <feltétel> záradék követheti.
- Ebben az esetben a feltétel az egyes csoportokra vonatkozik, ha egy csoport nem teljesíti a feltételt, nem lesz benne az eredményben.

A HAVING feltételére vonatkozó megszorítások

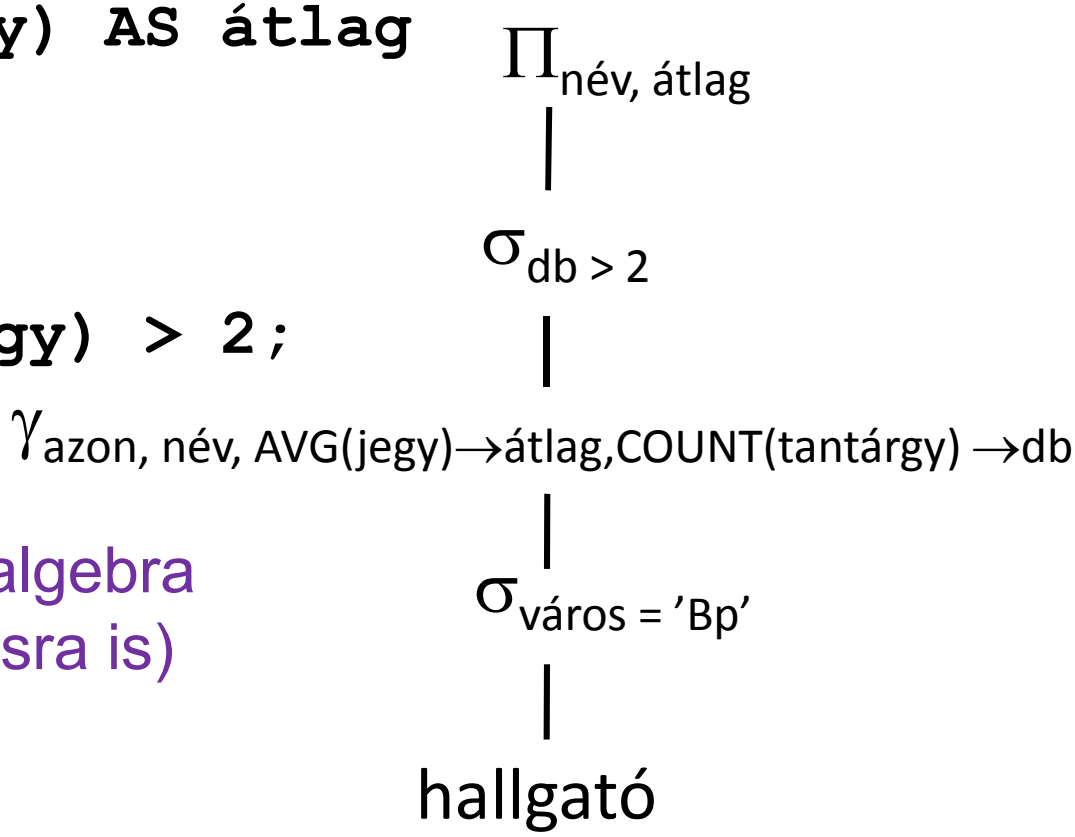
- Az alkérdésre nincs megszorítás.
- Az alkérdésen kívül csak olyan attribútumok szerepelhetnek, amelyek:
  1. vagy csoportosító attribútumok,
  2. vagy összesített attribútomok.

  (Azaz ugyanazok a szabályok érvényesek, mint a SELECT záradéknál).

Példa (egy relación) csoportosításra

Példa: hallgató (azon, név, város, tantárgy, jegy)

```
SELECT név, AVG(jegy) AS átlag
FROM hallgató
WHERE város = 'Bp'
GROUP BY azon, név
HAVING COUNT(tantárgy) > 2;
```



(Megjegyzés: a relációs algebra kibővítése a csoportosításra is)

Példa (több reláción) csoportosításra

```
SELECT onev, AVG(fizetes) + 100 emelt
FROM dolgozo d, osztaly o
WHERE d.oazon=o.oazon AND telephely='Bp'
GROUP BY o.oazon, onev
HAVING COUNT(dkod) > 3
ORDER BY onev;
```

$$\tau_{onev}(\pi_{onev,átlagfiz+100->emelt}(\sigma_{létszám>3}(\gamma_{o.oazon,onev,AVG(fizetes)->átlagfiz,COUNT(dkod)\ ->létszám}(\sigma_{d.oazon=o.oazon \wedge telephely='Bp'}(d \times o)))))$$

- Az operátorok egymás utáni alkalmazását **kifejezésfa** formájában is rajzolhatjuk fel!

Példa alkérdésre a HAVING-ben

- A Felszolgál(bár, sör, ár) és Sörök(név, gyártó) táblák felhasználásával adjuk meg az átlagos árát azon söröknek, melyeket
  - legalább három bárban felszolgálnak,
  - vagy Pete a gyártójuk.

Megoldás (de ennél jobb is van)

```
SELECT sör, AVG(ár)
FROM Felszolgál
GROUP BY sör
HAVING COUNT(bár) >= 3  OR
    sör IN (SELECT név
            FROM Sörök
            WHERE gyártó = 'Pete');
```

(HAVING…) Sör csoportok, Melyeket legalább három nem-NULL bárban árulnak, Vagy Pete a gyártójuk.

(SELECT…) Sörök, melyeket Pete gyárt (ez az Ullman mo., de

Itt jobb lenne más ez WHERE felt. & subs2 unio subs2

ORDER BY: Az eredmény rendezése

- SQL SELECT utasítás utolsó záradéka
- Az SQL lehetővé teszi, hogy a lekérdezés eredménye bizonyos sorrendben legyen rendezve. Az első attribútum egyenlősége esetén a 2.attribútum szerint rendezve, stb, minden attribútumra lehet növekvő vagy csökkenő sorrend.
- Select-From-Where utasításhoz a következő záradékot adjuk, a WHERE záradék és minden más záradék (mint például GROUP BY és HAVING) után következik:

  SELECT … FROM … [WHERE …] […]

  ORDER BY {attribútum [DESC], …}

- **Példa: SELECT * FROM Felszolgál**

  ORDER BY ár DESC, sör

Összefoglalva: záradékok

- Teljes SELECT utasítás

  (záradékok sorrendje nem cserélhető fel)

```
SELECT  [DISTINCT] …
FROM …
   [WHERE   … ]
   [GROUP BY …
      [HAVING … ] ]
   [ORDER BY …]
```