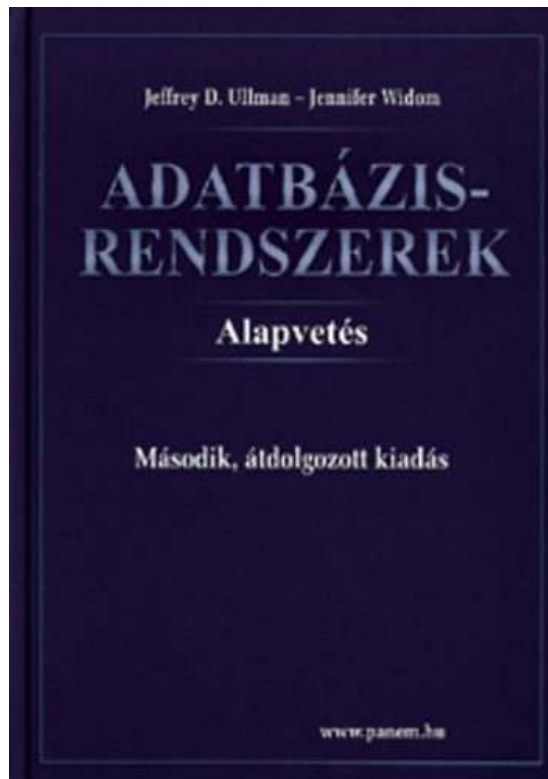


Lekérdezések az SQL-ben 3.rész



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

6.3. Alkérdezések, korrelált alkérdezések

Alkérdeések

- A FROM listán és a WHERE záradékban (valamint később lesz a HAVING záradékban) zárójelezett SELECT-FROM-WHERE utasításokat (alkérdeéseket) is használhatunk.
- Szintaktikus alakja: zárójelbe kell tenni a lekérdezést
- Hol használható? Ott, ahol relációnevet használunk:
 - (1) 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.

- (2) WHERE záradékban: kifejezésekben, feltételekben

Alkérdeések használata FROM listán I.

- **FROM záradékban** alkérdeéssel létrehozott ideiglenes táblát is megadhatunk. Ilyenkor a legtöbb esetben meg kell adnunk a sorváltozó nevét. **Szintaktikus alakja:**
(lekérdezés) [AS] sorváltozó
- **Szemantikája:** A FROM záradékban kiértékelődik az alkérdeés, utána a sorváltozót ugyanúgy használjuk, mint a közönséges adatbázis relációkat.
- **Példa:** Keressük meg a Joe's bár vendégei által kedvelt söröket (a feladatnak sok megoldása van)

Alkérdeések használata FROM listán II.

- **FROM záradékban** alkérdeéssel létrehozott ideiglenes táblát is megadhatunk. Ilyenkor a legtöbb esetben meg kell adnunk a sorváltozó nevét.
- **Példa:** Keressük meg a Joe's bár vendégei által kedvelt söröket.

SELECT sör

FROM Kedvel, (SELECT név

FROM Látogat

WHERE bár = 'Joe' 's bar') JD

WHERE Szeret.név = JD.név;

Sörivők, akik látogatják
Joe's bárját.



Alkérdeések használata 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 Θ [ANY | ALL] (lekérdezés)
- (iii) **Teljes, többdimenziós tábla** a visszatérő érték:
 - [NOT] EXISTS (lekérdezés)
 - (kif₁, ... kif_n) [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 I.

- 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 II.

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

SELECT bár

FROM Felszolgál

WHERE sör = 'Miller' AND

ár = (SELECT ár

FROM Felszolgál

WHERE bár = 'Joe' 's bar'

AND sör = 'Bud') ;

Ennyit kér
Joe a Budért.

Az IN művelet a WHERE záradékban

- sor IN (alkérdés) akkor és csak akkor **igaz**, ha a sor eleme az alkérdés eredményének.
 - 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 kedvel.

FROM Kedvel

WHERE név = 'Fred') ;

Tk.példa: Skalár értéket adó alkérdések

- Csillagok háborúja film gyártásirányítója:

```
SELECT név
```

```
FROM GyártásIrányító
```

```
WHERE azonosító =
```

```
  (SELECT producerAzon
```

```
    FROM Filmek
```

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

```
  );
```

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
    ) ;
```

Az 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.

Az ALL művelet

- $x \neq \text{ALL}(\text{alkérdés})$ akkor és csak akkor igaz, ha x az alkérdés egyetlen sorával sem egyezik meg.
- $\neq$ 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.

