

Adatbázisok 1.

SQL bevezetés – 1. rész

Select-From-Where záradékok

Több relációt tartalmazó lekérdezések

Alkérdezések

Miért az SQL?

- Az SQL magas szintű programozási nyelv.
 - A “hogyan” helyett “mit”
 - Így elkerülünk egy csomó macerát a procedurális nyelvekhez (pl. C++ vagy Java) képest

Miért az SQL?

- Az SQL magas szintű programozási nyelv.
 - A “hogyan” helyett “mit”
 - Így elkerülünk egy csomó macerát a procedurális nyelvekhez (pl. C++ vagy Java) képest
- Az adatbázis-kezelő rendszer kitalálja a leggyorsabb végrehajtási módot.
 - Ezt nevezik “lekérdezés optimalizációnak.”

Select-From-Where záradékok (*clauses*)

SELECT az érdekes attribútumok

FROM egy vagy több tábla

WHERE a táblák soraira vonatkozó feltételek

A példa, amit használunk

- Minden SQL lekérdezést (*SQL query*) a következő adatbázisséma fölött hajtunk végre.
 - Az aláhúzás a kulcsattribútumokat jelöli.

Teák(név, gyártó)

Teázók(név, cím, engedélySzám)

Vendégek(név, cím, telefon)

Szeret(vendég, tea)

Felhasználó(teázó, tea, ár)

Látogat(vendég, teázó)

Példa

Teák(név, gyártó)

Teázók(név, cím, engedélySzám)

Vendégek(név, cím, telefon)

Szeret(vendég, tea)

Felhasználó(teázó, tea, ár)

Látogat(vendég, teázó)

- A **Teák(név, gyártó)** táblában mely teákat gyártotta a Lipton?

```
SELECT név
```

```
FROM Teák
```

```
WHERE gyártó = 'Lipton';
```

Teák(név, gyártó)

Teázók(név, cím, engedélySzám)

Vendégek(név, cím, telefon)

Szeret(vendég, tea)

Felszolgál(teázó, tea, ár)

Látogat(vendég, teázó)

A lekérdezés eredménye

név
Yellow label
Brisk
Pyramid
...

Az eredmény egyetlen attribútumot (név) tartalmaz,
a sorokat Lipton által gyártott teák adják.

A lekérdezés jelentése

- Kezdjük a FROM záradékban megadott relációval.
- Alkalmazzuk a WHERE záradékban megadott kiválasztási feltételt.
- Levetítjük az eredményt a SELECT záradékban megadott oszlopokra.

Szemantika (a példában)

Teák(név, gyártó)

Teázók(név, cím, engedélySzám)

Vendégek(név, cím, telefon)

Szeret(vendég, tea)

Felszolgál(teázó, tea, ár)

Látogat(vendég, teázó)

név	gyártó
Brisk	Lipton

t.név bekerül az
eredménybe, ha igen.

Ellenőrizzük, hogy
Lipton-e.

A t sorváltozóval (*tuple variable*)
a sorokat vesszük
egymás után.

* a SELECT záradékban

- Ha egy reláció szerepel a FROM záradékban, a * SELECT záradékban a reláció összes attribútumát helyettesíti.
- Példa: Teák(név, gyártó):

```
SELECT *  
FROM Teák  
WHERE gyártó = 'Lipton';
```

A válasz:

név	gyártó
Yellow label	Lipton
Brisk	Lipton
Pyramid	Lipton
...	...

Azaz a **Teák** reláció összes attribútuma szerepel.

Attribútumok átnevezése

- Az attribútumok átnevezéséhez “AS <new name>” utasítást használhatjuk.
- Példa: Teák(név, gyártó):

```
SELECT név AS tea, gyártó  
FROM Teák  
WHERE gyártó = 'Lipton';
```

Az eredmény:

tea	gyártó
Yellow label	Lipton
Brisk	Lipton
Pyramid	Lipton
...	...

A SELECT záradék kifejezései

- Minden kifejezés, ami „értelmesnek tűnik” megjelenhet a SELECT záradékban.
- Példa: Felszolgál(teázó, tea, ár):

```
SELECT teázó, tea,  
        ár*114 AS árJenben  
FROM Felszolgál;
```

Az eredmény

teázó	tea	árJenben
Joe's	Brisk	285
Sue's	Pyramid	342
...

Konstansok

- Szeret(vendég, tea):

```
SELECT vendég,  
        'szereti a Brisket' AS BriskIvó  
FROM Szeret  
WHERE tea = 'Brisk';
```


Az eredmény

vendég	Brisklvó
Sally	szereti a Brisket
Fred	szereti a Brisket
...	...

Információ integráció (*information integration*)

- Sokszor az adatbázisokat sok forrásból (*sources*) építik fel (adattárházak).
- Tegyük fel, hogy minden teázónak van egy saját *Menü(tea, ár)* táblája.
- A *Felszolgál(teázó, tea, ár)* tábla elkészítéséhez minden ilyen táblát fel kell dolgoznunk és a teázó nevét konstansként kell beszúrnunk.

Információ integráció --- (2)

- Például Joe teázója esetében ezzel a lekérdezéssel dolgozhatunk:

```
SELECT 'Joe teázója' AS teázó, tea, ár  
FROM Menü;
```

Összetett feltételek (*conditions*) a WHERE záradékban

- Logikai műveletek (*boolean operators*): AND, OR, NOT.
- Összehasonlítások (*comparisons*) =, <>, <, >, <=, >=.

Példa összetett feltételre

Teák(név, gyártó)

Teázók(név, cím, engedélySzám)

Vendégek(név, cím, telefon)

Szeret(vendég, tea)

Felhasználó(teázó, tea, ár)

Látogat(vendég, teázó)

- A Felhasználó(teázó, tea, ár) táblában keressük meg Joe teázójában mennyit kérnek a Brisk teaért:

```
SELECT ár
```

```
FROM Felhasználó
```

```
WHERE teázó = 'Joe teázója' AND
```

```
tea = 'Brisk';
```

Minták (*patterns*)

- A feltételekben a sztringeket mintákra illeszthetjük
 - <Attribútum> LIKE <minta> vagy <Attribútum> NOT LIKE <minta>
- *Minta* aposztrófok közötti szöveg az alábbi jelekkel: % = “akármennyi karakter”; _ = “tetszőleges karakter, pontosan egy.”

Példa: LIKE

- Az **Vendégek(név, cím, telefon)** keressük a budapestieket.

```
SELECT név  
FROM Vendégek  
WHERE cím LIKE '%Budapest%';
```

Teák(név, gyártó)

Teázók(név, cím, engedélySzám)

Vendégek(név, cím, telefon)

Szeret(vendég, tea)

Felszolgál(teázó, tea, ár)

Látogat(vendég, teázó)

Példa: LIKE

- Az **Vendégek(név, cím, telefon)** keressük azokat a vendégeket, akiknek a nevük második karaktere ,a', és a nevük ,y'-ra végződik!

```
SELECT név  
FROM Vendégek  
WHERE név LIKE '_a%y' ;
```


NULL értékek

- A sorok mezői az SQL relációkban NULL értékeket is tartalmazhatnak.
- A jelentés a kontextustól függően változhat.
Általában:
 - *hiányzó érték* : pl. nem ismerjük Joe teázója címét.
 - *értelmetlen* : egy szingli esetében a házastárs neve.

NULL összehasonlítás

- Az SQL valójában háromértékű logikát (*three-valued logic*) használ: TRUE, FALSE, UNKNOWN.
- Ha egy értéket (NULL értéket is beleértve) NULL-lal hasonlítunk, az eredmény UNKNOWN.
- Egy sor akkor és csak akkor kerül be az eredménybe, ha a WHERE záradék TRUE értéket ad.

Háromértékű logika

- Tegyük fel a következőt: TRUE = 1, FALSE = 0, és UNKNOWN = $\frac{1}{2}$.
- Ekkor: AND = MIN; OR = MAX, NOT(x) = $1-x$.

- Példa:

TRUE AND (FALSE OR NOT(UNKNOWN)) = MIN(1, MAX(0, (1 - $\frac{1}{2}$))) = MIN(1, MAX(0, $\frac{1}{2}$)) = MIN(1, $\frac{1}{2}$) = $\frac{1}{2}$.

Teák(név, gyártó)
Teázók(név, cím, engedélySzám)
Vendégek(név, cím, telefon)
Szeret(vendég, tea)
Felszolgál(teázó, tea, ár)
Látogat(vendég, teázó)

Meglepetés!

- Az alábbi Felszolgál tábla esetén:

teázó	tea	ár
Joe teázója	Brisk	NULL

SELECT teázó

FROM Felszolgál

WHERE ár < 2.00 OR ár >= 2.00;

