

Adatbázisok 1.

SQL haladó – 2. rész

Külső összekapcsolások, Csoportosítás/Összesítés,
Beszúrás/Törlés/Módosítás,
Táblák létrehozása/Kulcs megszorítások

Adatbázis módosítások

- A *módosító* utasítások nem adnak vissza eredményt, mint a lekérdezések, hanem az adatbázis tartalmát változtatják meg.
- 3-féle módosító utasítás létezik:
 1. *Insert* (beszúrás).
 2. *Delete* (törlés).
 3. *Update* (létező sorok értékeinek módosítása).
- Data Manipulation Language (DML).

Beszúrás

Sörök(név, gyártó)

Kocsmák(név, cím, engedélySzám)

Alkeszek(név, cím, telefon)

Szeret(alkesz, sör)

Felhasznál(kocsma, sör, ár)

Látogat(alkesz, kocsma)

- Ha egyetlen sort szűrünk be:

```
INSERT INTO <reláció>
```

```
VALUES ( <attribútum lista> );
```

- **Példa:** a Szeret(alkesz, sör) táblában rögzítjük, hogy Susy szereti a Bud sört.

```
INSERT INTO Szeret
```

```
VALUES ( 'Susy' , 'Bud' );
```

Az INSERT-nél megadhatjuk az attribútumokat

- A reláció neve után megadhatjuk az attribútumait.
- Ennek két oka lehet:
 1. elfelejtettük, hogy a reláció definíciójában, milyen sorrendben szerepeltek az attribútumok.
 2. Nincs minden attribútumnak értéke, és azt szeretnénk, ha a hiányzó értékeket NULL vagy default értékkel helyettesítenék.

Példa: Attribútumok megadása

Sörök(név, gyártó)

Kocsák(név, cím, engedélySzám)

Alkerek(név, cím, telefon)

Szeret(alkes, sör)

Felolgal(kocsm, sör, ár)

Látogat(alkes, kocsm)

```
INSERT INTO Szeret (sör, alkesz)
VALUES ('Bud', 'Susy');
```

Default (alapértelmezett) értékek megadása

- A CREATE TABLE utasításban az oszlopnevet DEFAULT kulcsszó követheti és egy érték.
- Ha egy beszúrt sorban hiányzik az adott attribútum értéke, akkor a default értéket kapja.

Példa: Default (**alapértelmezett**) érték

Sörök(név, gyártó)

Kocsmák(név, cím, engedélySzám)

Alkeszek(név, cím, telefon)

Szeret(alkesz, sör)

Felhasznál(kocsma, sör, ár)

Látogat(alkesz, kocsma)

```
CREATE TABLE Alkeszek (  
    név CHAR(30) PRIMARY KEY,  
    cím CHAR(50) DEFAULT '123 Sesame St.',  
    telefon CHAR(16)  
);
```

Sörök(név, gyártó)
Kocsmák(név, cím, engedélySzám)
Alkeszek(név, cím, telefon)
Szeret(alkesz, sör)
Felszolgál(kocsma, sör, ár)
Látogat(alkesz, kocsma)

Példa: Default (alapértelmezett) értékek

```
INSERT INTO Alkeszek (név)
VALUES ( 'Susy' ) ;
```

Az eredmény sor:

név	cím	telefon
Susy	123 Sesame St	NULL

Több sor beszúrása

- Egy lekérdezés eredményét is beszúrhatjuk a következő módon:

```
INSERT INTO <reláció>  
( <alkérdés> );
```

Példa: Beszúrás alkérdéssel

Sörök(név, gyártó)

Kocsmák(név, cím, engedélySzám)

Alkeszek(név, cím, telefon)

Szeret(alkesz, sör)

Felhasznál(kocsmák, sör, ár)

Látogat(alkesz, kocsmák)

- A **Látogat(alkesz, kocsmák)** tábla felhasználásával adjuk hozzá a **Szesztestvérek(név)** táblához Susy szesztestvéreit, vagyis azokat, akikkel legalább egy közös kocsmát látogatnak.

Megoldás

A szesztestvér.

Alkesz párok: az első Susy, a második nem Susy, de van olyan kocsmá, amit mindketten látogatnak.

```
INSERT INTO Szesztestvérek
```

```
(SELECT I2.alkesz
```

```
FROM Látogat I1, Látogat I2  
WHERE I1.alkesz = 'Susy' AND  
I2.alkesz <> 'Susy' AND  
I1.kocsmá = I2.kocsmá  
);
```

Sörök(név, gyártó)

Kocsmák(név, cím, engedélySzám)

Alkeszek(név, cím, telefon)

Szeret(alkesz, sör)

Felhasznál(kocsmá, sör, ár)

Látogat(alkesz, kocsmá)

Törlés

- A törlendő sorokat egy feltétel segítségével adjuk meg:

```
DELETE FROM <reláció>  
WHERE <feltétel>;
```

Példa: Törlés

Sörök(név, gyártó)
Kocsmák(név, cím, engedélySzám)
Alkeszek(név, cím, telefon)
Szeret(alkesz, sör)
Felszolgál(kocsma, sör, ár)
Látogat(alkesz, kocsma)

```
DELETE FROM Szeret
WHERE alkesz = 'Susy' AND
      sör = 'Bud' ;
```

Példa: Az összes sor törlése

Sörök(név, gyártó)
Kocsmák(név, cím, engedélySzám)
Alkeszek(név, cím, telefon)
Szeret(alkesz, sör)
Felszolgál(kocsma, sör, ár)
Látogat(alkesz, kocsma)

```
DELETE FROM Szeret;
```

Példa: Több sor törlése

- A **Sörök**(név, gyártó) táblából töröljük azokat a söröket, amelyekhez létezik olyan sör, amit ugyanaz a cég gyártott.

Sörök(név, gyártó)

Kocsmák(név, cím, engedélySzám)

Alkeszek(név, cím, telefon)

Szeret(alkesz, sör)

Felszolgál(kocsma, sör, ár)

Látogat(alkesz, kocsma)

Példa: Több sor törlése

Sörök(név, gyártó)
Kocsmák(név, cím, engedélySzám)
Alkeszek(név, cím, telefon)
Szeret(alkesz, sör)
Felszolgál(kocsma, sör, ár)
Látogat(alkesz, kocsma)

- A Sörök(név, gyártó) táblából töröljük azokat a söröket, amelyekhez létezik olyan sör, amit ugyanaz a cég gyártott.

DELETE FROM Sörök s

WHERE EXISTS (

SELECT név FROM Sörök

WHERE gyártó = s.gyártó

AND név <> s.név);

Azok a sörök,
amelyeknek ugyanaz
a gyártója, mint az *s*
éppen aktuális
sorának, a nevük
viszont különböző.

A törlés szemantikája --- (1)

- Tegyük fel, hogy az Anheuser-Busch csak Bud és Bud Lite söröket gyárt.
- Tegyük fel még, hogy s sorai közt a Bud fordul elő először.
- Az alkérdés nem üres, a későbbi Bud Lite sor miatt, így a Bud törlődik.
- Kérdés, hogy a Bud Lite sor törlődik-e?

A törlés szemantikája --- (2)

- **Válasz:** igen, a Bud Lite sora is törlődik.
- A törlés ugyanis két lépésben hajtódik végre.
 1. Kijelöljük azokat a sorokat, amelyekre a WHERE feltétele teljesül.
 2. Majd töröljük a kijelölt sorokat.

Módosítás

- Bizonyos sorok bizonyos attribútumainak módosítása.

UPDATE <reláció>

SET <attribútum értékadások listája>

WHERE <sorokra vonatkozó feltétel>;

Példa: Módosítás

Sörök(név, gyártó)
Kocsmák(név, cím, engedélySzám)
Alkeszek(név, cím, telefon)
Szeret(alkesz, sör)
Felszolgál(kocsma, sör, ár)
Látogat(alkesz, kocsma)

- Frank telefonszámát 555-1212-re változtatjuk (Frank itt egy alkesz):

```
UPDATE Alkeszek  
SET telefon = '555-1212'  
WHERE név = 'Frank';
```

Példa: Több sor módosítása

Sörök(név, gyártó)
Kocsmák(név, cím, engedélySzám)
Alkeszek(név, cím, telefon)
Szeret(alkesz, sör)
Felszolgal(kocsma, sör, ár)
Látogat(alkesz, kocsma)

- Legfeljebb 4 dollárba kerülhessenek a sörök:

```
UPDATE Felszolgal  
SET ár = 4.00  
WHERE ár > 4.00;
```

Adatbázis sémák SQL-ben

- Data Definition Language (DDL), az SQL nyelv része, ennek segítségével hozhatunk létre adatobjektumokat, deklarálhatunk megszorításokat stb.

Relációk létrehozása

- A legegyszerűbb forma:

```
CREATE TABLE <név> (  
    <elemek listája>  
);
```

- Relációk törlése:

```
DROP TABLE <név>;
```

Tábla definíciók elemei

- A legegyszerűbb elem: az attribútum és annak típusa.
- A legfontosabb típusok a következők:
 - INT vagy INTEGER (szinonimák).
 - REAL vagy FLOAT (szinonimák).
 - CHAR(n) = rögzített hosszúságú sztring n karakter hosszú.
 - VARCHAR(n) = változó hosszúságú sztring legfeljebb n karakter hosszú.

Példa : Create Table

Sörök(név, gyártó)
Kocsmák(név, cím, engedélySzám)
Alkeszek(név, cím, telefon)
Szeret(alkesz, sör)
Felszolgál(kocsma, sör, ár)
Látogat(alkesz, kocsma)

```
CREATE TABLE Felszolgál (  
    kocsm    CHAR(20) ,  
    sör      VARCHAR(20) ,  
    ár       REAL  
);
```

SQL értékek

- Az egészek és lebegőpontos típusú konstansokat csak „szimplán le kell írni”. (Pont jelöli a tizedesvesszőt.)
- A sztringek esetében aposztrófok közé kell tennünk a konstansokat.
 - Két aposztróf = egyetlen aposztróf, **például**: ' Joe' ' s Bar' .
- Minden érték lehet NULL is.

Dátum és idő

- A DATE és TIME külön típusok az SQL-ben.
- A dátum típus formátuma:

DATE 'yyyy-mm-dd'

- Példa: DATE '2007-09-30' (2007. szeptember 30.)

Idő típus

- Az TIME típus formátuma:

TIME 'hh:mm:ss'

Optionálisan tizedespont is következhet a másodpercek után.

- **Példa:** TIME '15:30:02.5' = fél négy múlt két és fél másodperccel.

Kulcsok megadása (deklarálása)

- Egy attribútumot vagy attribútum listát kulcsként deklarálhatunk (PRIMARY KEY vagy UNIQUE).
- Mindkét formája a megszorításnak azt követeli meg, hogy relációnak ne legyen két olyan sora, melyek megegyeznek a kulcs attribútumokon.
- A különbségekről később lesz szó.

Egy attribútumos kulcs deklarációja

Sörök(név, gyártó)
Kocsmák(név, cím, engedélySzám)
Alkeszek(név, cím, telefon)
Szeret(alkesz, sör)
Felszolgál(kocsma, sör, ár)
Látogat(alkesz, kocsma)

- PRIMARY KEY vagy UNIQUE kulcsszót írhatjuk közvetlenül az attribútum mögé.
- Példa:

```
CREATE TABLE Sörök (  
    név          CHAR(20)  UNIQUE,  
    gyártó       CHAR(20)  
);
```

Több attribútumú kulcsok megadása

- A kulcs deklaráció a CREATE TABLE utasítás egy eleme is lehet az attribútum-típus elemek után.
- Több attribútumú kulcsokat csak ebben a formában deklarálhathatunk.
 - Ugyanakkor az egyetlen attribútumból álló kulcsokat is megadhatjuk ily módon.

Példa: Több attribútumú kulcs

Sörök(név, gyártó)

Kocsma(név, cím, engedélySzám)

Alkeszek(név, cím, telefon)

Szeret(alkesz, sör)

Felhasznál(kocsma, sör, ár)

Látogat(alkesz, kocsma)

```
CREATE TABLE Felhasznál (
    koccsma    CHAR(20),
    sör        VARCHAR(20),
    ár         REAL,
    PRIMARY KEY (koccsma, sör)
);
```


PRIMARY KEY vs. UNIQUE

1. Egy relációhoz egyetlen PRIMARY KEY tartozhat és több UNIQUE megszorítás.
2. A PRIMARY KEY egyetlen attribútuma sem kaphat NULL értéket. A UNIQUE megszorításnál szerepelhetnek NULL értékek egy soron belül akár több is.

SQL példák

- Tegyük fel, hogy az alábbi táblát hozták létre:

```
CREATE TABLE Hallgatók (  
    név CHAR(30) PRIMARY KEY,  
    cím CHAR(50)  
);
```

SQL példák

Lehet-e olyan hallgató, akinek nincs kitöltve (NULL) a neve?

Tegyük fel, hogy a név után mégsem szerepelne a PRIMARY KEY kulcsszó. Hány különböző nevű hallgató van?

```
SELECT COUNT(DISTINCT név) FROM Hallgatók;
```

Hogyan tudjuk meghatározni, hogy az azonos nevű hallgatókból hány darab van?

```
SELECT név, COUNT(név) FROM Hallgatók  
GROUP BY név  
HAVING COUNT(név) > 0;
```