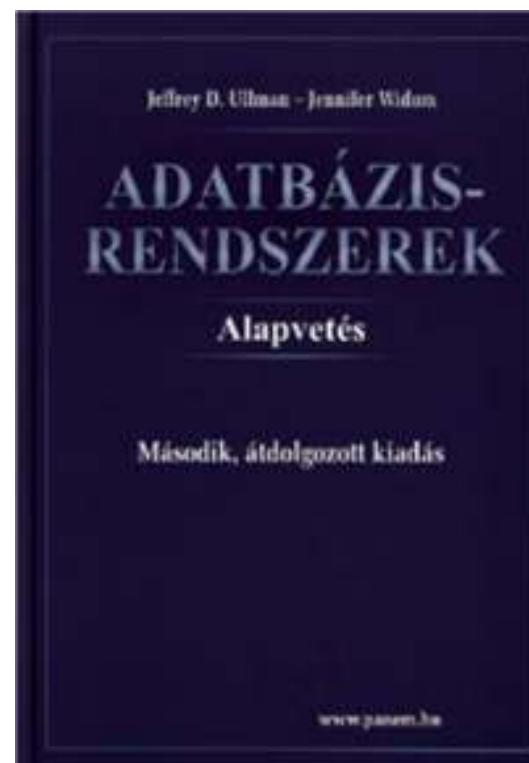


Az SQL adatbázisnyelv: DML

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

6.5. Az adatbázis tartalmának
módosítása (DML utasítások)
INSERT, DELETE, UPDATE



Adatbázis tartalmának módosítása

- **A módosító utasítások** nem adnak vissza eredményt, mint a lekérdezések, hanem az adatbázis tartalmát változtatják meg.
- 3-féle módosító utasítás létezik:
 - INSERT** - sorok beszúrása
 - DELETE** – sorok törlése
 - UPDATE** – sorok komponensei értékeinek módosítása

Beszúrás (insert into)

- Két alakja van: 1.) ha egyetlen sort szúrunk be:

```
INSERT INTO <reláció>
```

```
VALUES ( <konkrét értékek listája> );
```

- 2.) ha több sort, egy lekérdezés eredményét visszük fel alkérdés segítségével:

```
INSERT INTO <reláció>
```

```
( <alkérdés> );
```

- **Példa:** a **Kedvel(név, sör)** táblában rögzítjük, hogy Zsu szereti a Bud sört.

```
INSERT INTO Kedvel
```

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

Attribútumok megadása

- A reláció neve után megadhatjuk az attribútumait.
- Ennek alapvetően 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:

```
INSERT INTO Kedvel(sör, név)  
VALUES('Bud', 'Zsu');
```

Default 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.

```
CREATE TABLE Sörivók (  
    név CHAR(30) PRIMARY KEY,  
    cím CHAR(50) DEFAULT 'Sesame St'  
    telefon CHAR(16) );  
INSERT INTO Sörivók(név)  
VALUES ('Zsu');
```

Az eredmény sor:

név	cím	telefon
Zsu	Sesame St	NULL

Több sor beszúrása

- Egy lekérdezés eredményét is beszúrhatjuk:
INSERT INTO <reláció>
(<alkérdés>);
- A **Látogat(név, söröző)** tábla felhasználásával adjuk hozzá a **LehetBarát(név)** táblához Zsu „lehetséges barátait”, vagyis azokat a sörivőket, akik legalább egy olyan sörözőt látogatnak, ahova Zsu is szokott járni.
(megoldás a köv.oldalon)

Megoldás: Több sor beszúrása

(SELECT) a másik sörivó

```
INSERT INTO LehetBarát
(SELECT I2.név
FROM Látogat I1, Látogat I2
WHERE I1.név = 'Zsu' AND
      I2.név <> 'Zsu' AND
      I1.bár = I2.bár
);
```

(FROM) névpárok:
az első Zsu,
a második nem Zsu,
de van olyan bár,
amit mindketten
látogatnak.

Tk.Példa INSERT INTO utasításra

- A lekérdezést teljesen ki kell értékelni, mielőtt a sorokat beszúrnánk.
- Tankönyv 6.36 példa: új stúdiók beszúrása

```
INSERT INTO Stúdió (név)
(SELECT DISTINCT stúdióNév
FROM Filmek
WHERE stúdióNév NOT IN
(SELECT név FROM Stúdió));
```


Törlés (delete)

- A törlendő sorokat egy WHERE feltétel segítségével adjuk meg:

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

- Példa:

```
DELETE FROM Kedvel  
WHERE nev = 'Zsu' AND  
sör = 'Bud';
```

- Az összes sor törlése:

```
DELETE FROM Kedvel;
```

Példa: Több sor törlése

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

Példa: Több sor törlése

- 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);
```

(WHERE) azok a sörök, amelyeknek ugyanaz a gyártója, mint az s éppen aktuális sorának, a nevük viszont különböző.

A törlés szemantikája

- 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:** a Bud Lite sor törlődik-e?
(válasz a köv. oldalon)

A törlés szemantikája

- **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 (update)

- Bizonyos sorok bizonyos attribútumainak módosítása.

UPDATE <reláció>

SET <attribútum értékadások listája>

WHERE <sorokra vonatkozó feltétel>;

- Fecó telefonszámát 555-1212-re változtatjuk (Fecó itt egy sörivó neve):

UPDATE Sörivók

SET telefon = '555-1212'

WHERE név = 'Fecó';

Példa: Több sor módosítása

- Legfeljebb 4 dollárba kerülhessenek a sörök:

```
UPDATE Felszolgál
```

```
SET ár = 4.00
```

```
WHERE ár > 4.00;
```

- Olcsó sörök árát duplázzuk

```
UPDATE Felszolgál
```

```
SET ár = 2 * ár
```

```
WHERE ár < 1.00;
```

Tk.Példa UPDATE utasításra

- Tankönyv 6.39 példa:

UPDATE GyártásIrányító

SET név = 'Ig.' || név

WHERE azonosító IN

(SELECT elnökAzon FROM Stúdió)

Tranzakciók

- **Tranzakció** = olyan folyamat, ami adatbázis lekérdezéseket, módosításokat tartalmaz.
- Az utasítások egy „értelmes egészt” alkotnak.
- Egyetlen utasítást tartalmaznak, vagy az SQL-ben explicit módon megadhatóak.

ACID tranzakciók

- **Az ACID tranzakciók:**
 - **Atomiság (atomicity):** vagy az összes vagy egy utasítás sem hajtódik végre.
 - **Konzisztencia (consistency):** az adatbázis megszorítások megőrződnek.
 - **Elkülönítés (isolation):** a felhasználók számára úgy tűnik, mintha folyamatok, elkülönítve, egymás után futnának le.
 - **Tartósság (durability):** egy befejeződött tranzakció módosításai nem vesznek el.
- **Opcionálisan:** gyengébb feltételek is megadhatóak.

COMMIT és ROLLBACK

- **A COMMIT utasítás** a tranzakció sikeres befejeződését eredményezi. Egy sikeresen befejeződött tranzakció a kezdete óta végrehajtott utasításainak módosításait tartósan rögzíti az adatbázisban
 - vagyis a módosítások *véglegesítődnek*.
- **A ROLLBACK utasítás** megszakítja a tranzakció végrehajtását, és annak sikertelen befejeződését eredményezi. Az így befejezett tranzakció SQL utasításai által végrehajtott módosításokat a rendszer meg nem történtekké teszi
 - Vagyis az összes utasítás *visszagörgetésre kerül*, a módosítások nem jelennek meg az adatbázisban.