

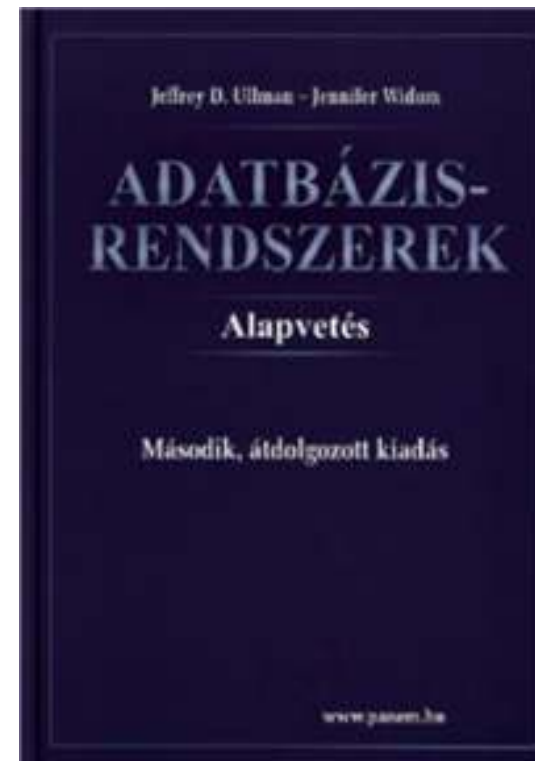
SQL DDL-2 (aktív elemek) triggerek

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

7.fej.: Megszorítások és triggerek

7.4. Önálló megszorítások

7.5. Triggerek



7.fejezet: Megszorítások és triggerek

- **Aktív elemek** – olyan kifejezés vagy utasítás, amit egyszer eltároltunk az adatbázisban és az azt várjuk tőle, hogy a megfelelő pillanatban lefusson (pl. adatok helyességének ellenőrzése)
- **A megszorítás** adatelemek közötti kapcsolat, amelyet az adatbázis-kezelő rendszernek fent kell tartania.
 - **Példa:** kulcs megszorítások.
- **Triggerek** olyankor hajtódnak végre, amikor valamilyen megadott esemény történik, mint például sorok beszúrása egy táblába.

Önálló megszorítások: Assertions

- SQL aktív elemek közül a leghatékonyabbak nincs hozzárendelve sem sorokhoz, sem azok komponenseihez, hanem **táblához kötődnek**.
- Ezek is **az adatbázissémához tartoznak** a relációsémákhoz és nézetekhez hasonlóan.
- **CREATE ASSERTION** <név>
CHECK (<feltétel>);
- A feltétel tetszőleges táblára és oszlopra hivatkozhat az adatbázissémából.


Példa: önálló megszorítások

- A Felszolgál(söröző, sör, ár) táblában nem lehet olyan söröző, ahol a sörök átlagára 5 dollárnál több

- **CREATE ASSERTION CsakOlcsó CHECK**

```
(  
  NOT EXISTS (  
    SELECT söröző  
    FROM Felszolgál  
    GROUP BY bár  
    HAVING 5.00 < AVG(ár)  
  ));
```

(SELECT ..
olyan sörözők,
ahol a sörök
átlagosan
drágábbak
5 dollárnál)



Példa: önálló megszorítások

- Az Sörvívó(név, cím, telefon) és Söröző(név, cím, engedély) táblákban nem lehet több bár, mint amennyi sörívó van.

```
CREATE ASSERTION KevésBár CHECK (  
    (SELECT COUNT(*) FROM Söröző) <=  
    (SELECT COUNT(*) FROM Sörívó)  
);
```

Önálló megszorítások ellenőrzése

- Alapvetően az adatbázis bármely módosítása előtt ellenőrizni kell.
- Egy okos rendszer felismeri, hogy mely változtatások, mely megszorításokat érinthetnek.
 - **Példa:** a **Sörök** tábla változásai nincsenek hatással az iménti KevésBár megszorításra. Ugyanez igaz a **Sörivók** táblába történő beszúrásokra is.

Tankönyv példája

Önálló megszorítás, amelyik a stúdióelnökök gazdagságát írja elő

```
CREATE ASSERTION GazdagElnök CHECK
(NOT EXISTS
  (SELECT *
   FROM Stúdió, GyártásIrányító
   WHERE elnökAzon = azonosító
   AND nettóBevétel < 10000000
  )
);
```

Miért hasznosak a triggererek?

- **Az önálló megszorításokkal** (assertions) sok mindent le tudunk írni, az ellenőrzésük azonban gondot jelenthet.
- **Az attribútumokra és sorokra vonatkozó megszorítások** ellenőrzése egyszerűbb (tudjuk mikor történik), ám ezekkel nem tudunk minden kifejezni.
- **A triggererek** esetén a felhasználó mondja meg, hogy egy megszorítás mikor kerüljön ellenőrzésre.

Esemény-Feltétel-Tevékenység szabályok

- A triggereket esetenként *ECA szabályoknak* (*event-condition-action*) *esemény-feltétel-tevékenység* szabályoknak is nevezik.
- **Esemény:** általában valamilyen módosítás a adatbázisban, INSERT, DELETE, UPDATE.
- **Mikor?:** BEFORE, AFTER, INSTEAD
- **Mit?:** OLD ROW, NEW ROW FOR EACH ROW
OLD/NEW TABLE FOR EACH STATEMENT
- **Feltétel :** SQL igaz-hamis-ismeretlen feltétel.
- **Tevékenység :** SQL utasítás, BEGIN..END,
PSM tárolt eljárás

Előzetes példa egy triggerre

- Ahelyett, hogy visszautasítanánk a **Felhasználó(söröző, sör, ár)** táblába történő beszúrást az ismeretlen sörök esetén, a **Sörök(név, gyártó)** táblába is beszúrjuk a megfelelő sort a gyártónak NULL értéket adva.

Példa: trigger definíció

```
CREATE TRIGGER SörTrig
  AFTER INSERT ON Felszolgál
  REFERENCING NEW ROW AS ÚjSor
  FOR EACH ROW
  WHEN (ÚjSor.sör NOT IN
        (SELECT név FROM Sörök))
  INSERT INTO Sörök(név)
    VALUES(ÚjSor.sör);
```

Esemény

Feltétel

Tevékenység

Triggerek --- 1

- A *triggerek*, amelyeket szokás *esemény-feltétel-tevékenység* szabályoknak is nevezni, az eddigi megszorításoktól három dologban térnek el:
- A triggeret a rendszer csak akkor ellenőrzi, ha bizonyos *események* bekövetkeznek.
A megengedett események általában egy adott relációra vonatkozó beszúrás, törlés, módosítás, vagy a tranzakció befejeződése.

Triggerek --- 2

- A kiváltó esemény azonnali megakadályozása helyett a trigger először egy *feltételt* vizsgál meg
- Ha a trigger feltétele teljesül, akkor a rendszer végrehajtja a triggerhez tartozó *tevékenységet*. Ez a művelet ezután megakadályozhatja a kiváltó esemény megtörténtét, vagy meg nem történtté teheti azt.

Tankönyv példája (7.5. ábra)

-- Nem engedi csökkenteni a gyártásirányítók nettó bevételét:

```
CREATE TRIGGER NetBevétTrigger
AFTER UPDATE OF nettóBevétel ON GyártásIrányító
REFERENCING
    OLD ROW AS RégiSor,
    NEW ROW AS ÚjSor
FOR EACH ROW
WHEN(RégiSor.nettóBevétel > ÚjSor.nettóBevétel)
UPDATE GyártásIrányító
SET nettóBevétel = RégiSor.nettóBevétel
WHERE azonosító = ÚjSor.azonosító;
```

Tankönyv példája (7.6. ábra)

-- Az átlagos nettó bevétel megszorítása:

```
CREATE TRIGGER ÁtlagNetBevétTrigger
AFTER UPDATE OF nettóBevétel ON GyártásIrányító
REFERENCING
    OLD TABLE AS RégiAdat,
    NEW TABLE AS ÚjAdat
FOR EACH STATEMENT
WHEN(500000 > (SELECT AVG(nettóBevétel)
                FROM GyártásIrányító)
DELETE FROM GyártásIrányító
WHERE (név, cím, azonosító) IN ÚjAdat;
INSERT INTO gyártásIrányító (SELECT * FROM ...)
```