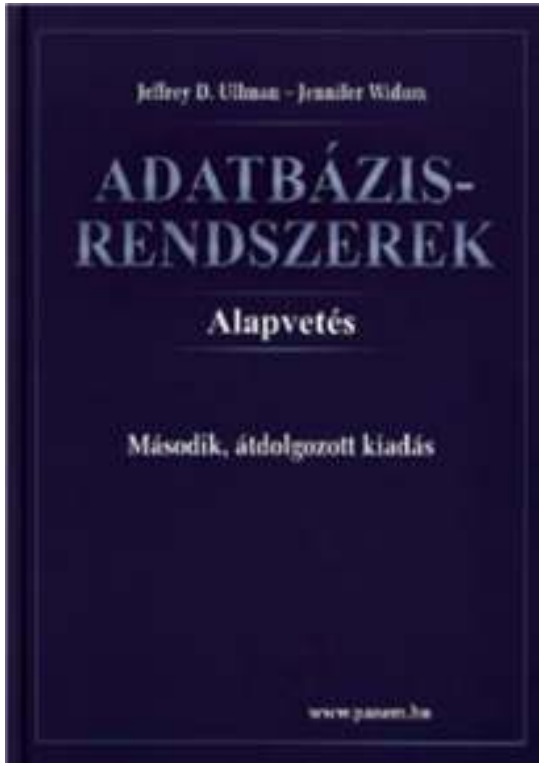
SQL DDL-1: táblák és megszorítások

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



2.3. Relációsémák definiálása
7.1. Kulcsok és idegen kulcsok
7.2. Értékekre és sorokra vonatkozó megszorítások
7.3. Megszorítások módosítása

--- folyt.(DDL-2) 7.4-7.5. Önálló megszorítások és triggerek

Adatbázis relációsémák definiálása

- Tankönyv 2.3. fejezete
- Az SQL tartalmaz adatleíró részt (DDL) is, az adatbázis objektumainak a leírására és megváltoztatására.
  Objektumok leíró parancsa a CREATE utasítás.
- Objektumok, például tábla, nézettábla, indextábla, stb.
- A relációt az SQL-ben táblának (TABLE) nevezik,
  az SQL alapvetően háromféle táblát kezel:
  - Alaptáblák (permanens) CREATE TABLE
  - Nézettáblák CREATE VIEW (ezt nem feltétlen tárolja)
  - Átmeneti munkatáblák WITH utasítás (lehet rekurzió is)
- Alaptáblák létrehozása: CREATE TABLE (köv.oldal)

Tábla/reláció sémák SQL-ben

- A legegyszerűbb formája:

```
CREATE TABLE relációnév (
        Attribútum deklarációk listája,
        További kiegészítések
);
```

- Az attribútum deklaráció legalapvetőbb elemei:

  Attribútumnév típus [kiegészítő lehetőségek]

- Itt: A **típus** olyan, amit az SQL konkrét megvalósítása támogat (gyakorlaton Oracle környezetben nézzük meg), Típusok, pl: INTEGER, REAL, CHAR, VARCHAR, DATE

- A kiegészítő lehetőségek például [PRIMARY KEY] vagy [DEFAULT érték] (köv.lapon példa)

Példa: sörivók adatbázis

Sörök(<u>név</u>, gyártó)
Sörözők(<u>név</u>, város, tulaj, engedély)
Sörivók(<u>név</u>, város, tel)

Kedvel(<u>név</u>, <u>sör</u>)
Felszolgál(<u>söröző</u>, <u>sör</u>, ár)
Látogat(<u>név</u>, <u>söröző</u>)

- Az aláhúzás jelöli a **kulcsot** (a sorok a kulcs összes attribútumán nem vehetik fel ugyanazt az értékeket).

  -- Ez a kulcs, külső kulcs és hivatkozási épség megszorításoknak lesz később kiváló példája.

Egyszerű példák táblák létrehozására

```
CREATE TABLE Sörözők (
    név CHAR(20),
    város VARCHAR2(40),
    tulaj CHAR(30),
    engedély DATE  DEFAULT SYSDATE
    );

CREATE TABLE Felszolgál (
          söröző CHAR(20),
          sör   VARCHAR2(20),
          ár   NUMBER(10,2) DEFAULT 1.00
    );
```

Az SQL értékekről (bővebben gyakorlaton)

- INTEGER, REAL, stb, a szokásos értékek, számok.
- STRING szintén, de itt egyes-aposztróf közé kell tenni a ‘szöveget’ (vagyis nem „macskaköröm” közé).
  - Két egyes-aposztróf = egynek felel meg, például `'Joe''s Bar'` megfelel a Joe’s Bar szövegnek.
- Bármely érték lehet NULL
- DATE és TIME típusok is vannak az SQL-ben.
- A dátum formátumát meg kell adni DATE ’yyyy-mm-dd’
  - Például: `DATE '2007-09-30'` (2007. szept. 30)
- Az idő formátumát is meg kell adni TIME ’hh:mm:ss’
  - Például: `TIME '15:30:02.5'` (délután fél 4 múlt két és fél másodperccel)

Kulcs megadása

- PRIMARY KEY vagy UNIQUE
- Nincs a relációnak két olyan sora, amely a lista minden attribútumán megegyezne.
- Kulcsoknál nincs értelme a DEFAULT értéknek.
- Kulcsok megadásának két változata van:
  - Egyszerű kulcs (egy attribútum) vagy
  - Összetett kulcs (attribútumok listája)

(példákat lásd a következő oldalon)

Egyszerű kulcs megadása

- Ha a kulcs egyetlen attribútum, akkor ez az attribútum deklarációban megadható

  <attribútumnév> <típus> PRIMARY KEY

  vagy <attribútumnév> <típus> UNIQUE

- Példa:

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

Összetett kulcs megadása

- Ha a kulcs több attribútumból áll, akkor a CREATE TABLE utasításban az attribútum deklaráció után a kiegészítő részben meg lehet adni további tábla elemeket: PRIMARY KEY (attrnév$_1$, … attrnév$_k$)
- Példa:

```
CREATE TABLE Felszolgál (
      söröző     CHAR(20),
      sör        VARCHAR2(20),
      ár         NUMBER(10,2),
      PRIMARY KEY (söröző, sör)
);
```

PRIMARY KEY vs. UNIQUE

- Csak egyetlen PRIMARY KEY lehet a relációban, viszont UNIQUE több is lehet.
- PRIMARY KEY egyik attribútuma sem lehet NULL érték egyik sorban sem. Viszont UNIQUE-nak deklarált attribútum esetén a táblának lehet olyan sora, ahol a UNIQUE attribútum értéke NULL

DDL – adatleíró részben módosítás

- Hogyan tudjuk a leíró részt módosítani?
  - CREATE – létrehozni
  - DROP – eldobni, a teljes leírást és mindazt, ami ehhez kapcsolódott hozzáférhetetlenné válik
  - ALTER – módosítani a leírást
- Ha ezt táblára használjuk
  - DROP TABLE táblanév;
  - ALTER TABLE táblanév

    DROP attribútumnév - - oszlopot tudunk törölni
    ADD attribútumnév értéktípus - - új oszlopot adni
    kiegészítő részek például megszorítások
- Például mikor adhatunk meg UNIQUE feltételt?

Megszorítások és triggerek

- Tankönyv 7. fejezet
- 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 AB rendszernek fent kell tartania.
  - Példa: kulcs megszorítások.
- Triggerek olyankor hajtódnak végre, amikor valamilyen megadott esemény történik, mint például sorok beszúrása egy táblába.

Megszorítások (áttekintés)

(1) Kulcsok és idegen kulcsok megadása

- A hivatkozási épség fenntartása
- Megszorítások ellenőrzésének késleltetése

(2) Értékekre vonatkozó feltételek

- NOT NULL feltételek
- Attribútumra vonatkozó CHECK feltételek

(3) Sorokra vonatkozó megszorítások

- Sorra vonatkozó CHECK feltételek

(4) Megszorítások módosítása (constraints)

(folyt.köv.) Önálló megszorítások, triggerek

Idegen kulcsok megadása

- Még egy kiegészítő lehetőség Mi köthet össze két táblát? Idegen kulcs (foreign key) megadása
- Az egyik tábla egyik oszlopában szereplő értékeknek szerepelnie kell egy másik tábla bizonyos attribútumának az értékei között.
- A hivatkozott attribútumoknak a másik táblában kulcsnak kell lennie! (PRIMARY KEY vagy UNIQUE)
- Példa: **Felszolgál(söröző, sör, ár)** táblára megszorítás, hogy a sör oszlopában szereplő értékek szerepeljenek a **Sörök(név, gyártó)** táblában a név oszlop értékei között.

Idegen kulcs megadása: attribútumként

REFERENCES kulcsszó használatának két lehetősége: attribútumként vagy sémaelemként lehet megadni.

1.) Attribútumonként (egy attribútumból álló kulcsra)

Példa:

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

CREATE TABLE Felszolgál (
    söröző CHAR(20),
    sör   CHAR(20) REFERENCES Sörök(név),
    ár    REAL );
```

Idegen kulcs megadása: sémaelemként

2.) Sémaelemként (egy vagy több attr.-ból álló kulcsra)

FOREIGN KEY (attribútum lista)

REFERENCES relációnév (attribútum lista)

Példa:

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

CREATE TABLE Felszolgál (
  söröző    CHAR(20),
  sör       CHAR(20),
  ár        REAL,
  FOREIGN KEY(sör) REFERENCES Sörök(név));
```

Idegen kulcs megszorítások megőrzése

- Példa: *R* = Felszolgál, *S* = Sörök.
- Egy idegen kulcs megszorítás *R* relációról *S* relációra kétféleképpen sérülhet:
  1. Egy *R*-be történő beszúrásnál vagy *R*-ben történő módosításnál *S*-ben nem szereplő értéket adunk meg.
  2. Egy *S*-beli törlés vagy módosítás „lógó” sorokat eredményez *R*-ben.

Hogyan védekezzünk? --- (1)

- Példa: *R* = Felszolgál, *S* = Sörök.
- Nem engedjük, hogy Felszolgál táblába a Sörök táblában nem szereplő sört szúrjanak be vagy Sörök táblában nem szereplő sörre módosítsák (nincs választási lehetőségünk, a rendszer visszautasítja a megszorítást sértő utasítást)
- A Sörök táblából való törlés vagy módosítás, ami a Felszolgál tábla sorait is érintheti (mert sérül az idegen kulcs megszorítás) 3-féle módon kezelhető (lásd köv.oldal)

Hogyan védekezzünk? --- (2)

1. Alapértelmezés (Default) : a rendszer nem hajtja végre a törlést.
2. Továbbgyűrűzés (Cascade): a Felszolgál tábla értékeit igazítjuk a változáshoz.
   - Sör törlése: töröljük a Felszolgál tábla megfelelő sorait.
   - Sör módosítása: a Felszolgál táblában is változik az érték.
3. Set NULL: a sör értékét állítsuk NULL-ra az érintett sorokban.

Példa: továbbgyűrűzés

- Töröljük a Bud sort a Sörök táblából:
  - az összes sort töröljük a Felszolgál táblából, ahol sör oszlop értéke ’Bud’.
- A ’Bud’ nevet ’Budweiser’-re változtatjuk:
  - a Felszolgál tábla soraiban is végrehajtjuk ugyanezt a változtatást.

Példa: Set NULL

- A Bud sort töröljük a Sörök táblából:
  - a Felszolgál tábla sör = 'Bud' soraiban a Budot cseréljük NULL-ra.
- 'Bud'-ról 'Budweiser'-re módosítunk:
  - ugyanazt kell tennünk, mint törléskor.

A stratégia kiválasztása

- Ha egy idegen kulcsot deklarálunk megadhatjuk a SET NULL és a CASCADE stratégiát is beszúrásra és törlésre is egyaránt.
- Az idegen kulcs deklarálása után ezt kell írnunk:

  ON [UPDATE, DELETE][SET NULL CASCADE]
- Ha ezt nem adjuk meg, a default stratégia működik.

Példa: stratégia beállítása

```
CREATE TABLE Felszolgál (
    söröző CHAR(20),
    sör      CHAR(20),
    ár        REAL,
    FOREIGN KEY(sör)
        REFERENCES Sörök(név)
        ON DELETE SET NULL
        ON UPDATE CASCADE
);
```

Megszorítások ellenőrzésének késleltetése

- Körkörös megszorítások miatt szükség lehet arra, hogy a megszorításokat ne ellenőrizze, amíg az egész tranzakció be nem fejeződött.
- Bármelyik megszorítás deklarálható DEFERRABLE (késleltethető) vagy NOT DEFERRABLE-ként (vagyis minden adatbázis módosításkor a megszorítás közvetlenül utána ellenőrzésre kerül). DEFERRABLE-ként deklaráljuk, akkor lehetőségünk van arra, hogy a megszorítás ellenőrzésével várjon a rendszer a tranzakció végéig.
- Ha egy megszorítás késleltethető, akkor lehet
  - INITIALLY DEFERRED (az ellenőrzés a tranzakció jóváhagyásáig késleltetve lesz) vagy
  - INITIALLY IMMEDIATE (minden utasítás után ellenőrzi)

Megszorítások (áttekintés)

(1) Kulcsok és idegen kulcsok

- A hivatkozási épség fenntartása
- Megszorítások ellenőrzésének késleltetése

(2) Értékekre vonatkozó feltételek

- NOT NULL feltételek
- Attribútumra vonatkozó CHECK feltételek

(3) Sorokra vonatkozó megszorítások

- Sorra vonatkozó CHECK feltételek

(4) Megszorítások módosítása (constraints)

(folyt.köv.) Önálló megszorítások, triggerek

Értékekre vonatkozó feltételek

- Egy adott oszlop értékeire vonatkozóan adhatunk meg megszorításokat.
- A CREATE TABLE utasításban az attribútum deklarációban NOT NULL kulcsszóval
- az attribútum deklarációban CHECK(<feltétel>)
  A feltétel, mint a WHERE feltétel, alkérdés is használható. A feltételben csak az adott attribútum neve szerepelhet, más attribútumok (más relációk attribútumai is) csak alkérdésben szerepelhetnek.

Példa: értékekre vonatkozó feltétel

```
CREATE TABLE Felszolgál (
    söröző CHAR(20) NOT NULL,
    sör      CHAR(20) CHECK ( sör IN
                  (SELECT név FROM Sörök)),
    ár       REAL CHECK ( ár <= 5.00 )
);
```

Mikor ellenőrzi?

- Érték-alapú ellenőrzést csak beszúrásnál és módosításnál hajt végre a rendszer.
  - Példa: `CHECK (ár <= 5.00)` a beszúrt vagy módosított sor értéke nagyobb 5, a rendszer nem hajtja végre az utasítást.
  - Példa: `CHECK (sör IN (SELECT név FROM Sörök))`, ha a Sörök táblából törlünk, ezt a feltételt nem ellenőrzi a rendszer.

Megszorítások (áttekintés)

(1) Kulcsok és idegen kulcsok

- A hivatkozási épség fenntartása
- Megszorítások ellenőrzésének késleltetése

(2) Értékekre vonatkozó feltételek

- NOT NULL feltételek
- Attribútumra vonatkozó CHECK feltételek

(3) Sorokra vonatkozó megszorítások

- Sorra vonatkozó CHECK feltételek

(4) Megszorítások módosítása (constraints)

(folyt.köv.) Önálló megszorítások, triggerek

Sorokra vonatkozó megszorítások

- A CHECK (<feltétel>) megszorítás a séma elemeként is megadható.
- A feltételben tetszőleges oszlop és reláció szerepelhet.
  - De más relációk attribútumai csak alkérdésben jelenhetnek meg.
- Csak beszúrásnál és módosításnál ellenőrzi a rendszer.

Példa: sor-alapú megszorítások

- Csak Joe bárja nevű söröző̋ben lehetnek drágábbak a sörök 5 dollárnál:

```
CREATE TABLE Felszolgál (
    söröző  CHAR(20),
    sör     CHAR(20),
    ár      REAL,
    CHECK (söröző= 'Joe bárja' OR ár <= 5.00)
    );
```

Tankönyv példája

Attribútumokra és sorokra vonatkozó megszorítások

Példa: Ha egy színész neme férfi, akkor

a neve nem kezdődhet ‘Ms.’-el

```
CREATE TABLE FilmSzínész (
    név CHAR(30) PRIMARY KEY,
    cím VARCHAR(255) NOT NULL,
    nem CHAR(1),
    születésiDátum DATE,
    CHECK (nem = 'N' OR név NOT LIKE 'Ms.%')
    );
```

Megszorítások (áttekintés)

(1) Kulcsok és idegen kulcsok

- A hivatkozási épség fenntartása
- Megszorítások ellenőrzésének késleltetése

(2) Értékekre vonatkozó feltételek

- NOT NULL feltételek
- Attribútumra vonatkozó CHECK feltételek

(3) Sorokra vonatkozó megszorítások

- Sorra vonatkozó CHECK feltételek

(4) Megszorítások módosítása (constraints)

(folyt.köv.) Önálló megszorítások, triggerek

Megszorítások elnevezése

- Nevet tudunk adni a megszorításoknak, amire később tudunk hivatkozni (könnyebben lehet törölni, módosítani)

Tankönyv példái:

- név CHAR(30) CONSTRAINT NévKulcs PRIMARY KEY,
- nem CHAR(1) CONSTRAINT FérfiVagyNő
  CHECK (nem IN ('F', 'N')),
- CONSTRAINT Titulus CHECK (nem = 'N' OR
  név NOT LIKE 'Ms.\%')

Megszorítások módosítása

Tankönyv példái:

- ALTER TABLE FilmSzínész ADD CONSTRAINT NévKulcs PRIMARY KEY (név);
- ALTER TABLE FilmSzínész ADD CONSTRAINT FérfiVagyNő CHECK (nem IN ('F', 'N'));
- ALTER TABLE FilmSzínész ADD CONSTRAINT Titulus CHECK (nem = 'N' OR név NOT LIKE 'Ms.\%');