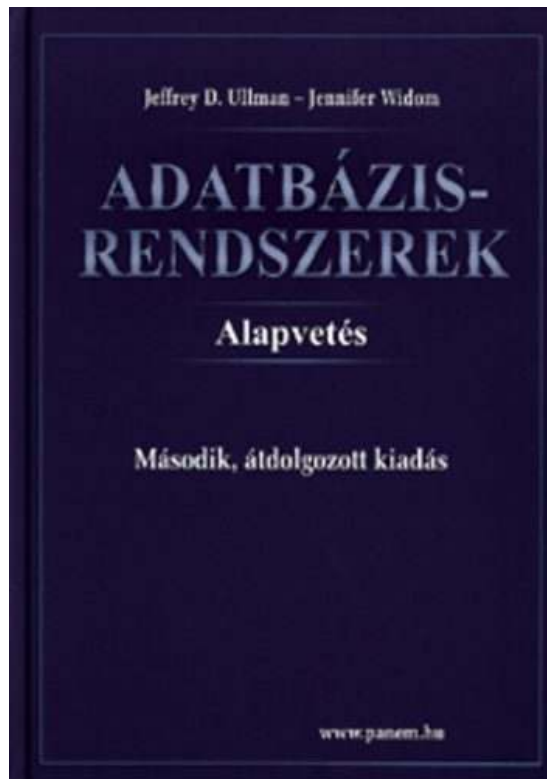


Nézetek és indexek



Ullman-Widom: Adatbázisrendszerek
Alapvetés

Második, átdolgozott kiadás, Panem,
2009

- 8.1. Nézetablák
- 8.2. Adatok módosítása
nézetablákon keresztül
- 8.3. Indexek az SQL-ben
- 8.4. Indexek kiválasztása
- 8.5. Tárolt nézetablák

Nézettáblák

- A *nézettábla* olyan reláció, amit tárolt táblák (*alaptáblák*) és más nézettáblák felhasználásával definiálunk.
- Kétféle létezik:
 1. *virtuális* = nem tárolódik az adatbázisban; csak a relációt megadó lekérdezés.
 2. *Materializált* = kiszámítódik, majd tárolásra kerül.

Nézettáblák létrehozása

- Deklaráció:

```
CREATE [MATERIALIZED] VIEW <név>  
AS <lekérdezés>;
```

- Alapesetben virtualizált nézettábla jön létre.

Példa: nézettábla definíció

- **Ihatja(név, sör)** nézettáblában a sörivők mellett azon söröket tároljuk, melyeket legalább egy bárban felszolgálnak az általa látogatottak közül:

```
CREATE VIEW Ihatja AS  
  SELECT név, sör  
  FROM Látogat, Felszolgál  
  WHERE L.bár = F.bár;
```

Példa: nézettáblákhoz való hozzáférés

- A nézettáblák ugyanúgy kérdezhetők le, mint az alaptáblák.
 - A nézettáblákon keresztül az alaptáblák néhány esetben módosíthatóak is, ha a rendszer a módosításokat át tudja vezetni.
- **Példa lekérdezés:**

```
SELECT sör FROM Ihatja  
WHERE név = 'Sally' ;
```

Materializált nézet táblák

- **Probléma:** minden alkalommal, amikor az alaptáblák valamelyike változik, a materializált nézet tábla frissítése is szükségessé válhat.
 - Ez viszont néha túl költséges.
- **Megoldás:** Periodikus frissítése a materializált nézet tábláknak, amelyek egyébként „nem aktuálisak”.

Példa: levelezési lista

- A következő levelezési lista **cs145-aut0708** valójában egy materializált nézet tábla, ami a kurzusra beiratkozott hallgatókat tartalmazza.
- Ezt négyszer frissítik egy nap.
 - A feliratkozás után közvetlen még nem feltétlen kapja meg az ember az akkor küldött emaileket.

Példa: adattárház

- Wal-Mart minden áruházának minden eladását egy adatbázisban tárolja.
- Éjszaka az új adatokkal frissítik az áruház lánc *adattárházát*, ami itt az eladások materializált nézeteiből áll.
- Az adattárházat aztán elemzők használják, hogy trendeket figyeljenek meg és odamozgassák az árukat, ahol azok a legjobb áron értékesíthetők.

Indexek

- *Index* = olyan adatszerkezet, amivel egy-egy reláció sorait gyorsabban érhetjük el adott attribútumának, attribútumainak értéke, értékei alapján.
- Lehet hash tábla, de az ab rendszerekben a legtöbb esetben kiegyensúlyozott keresési fával valósítják meg (*B-fák*).

Indexek deklarációja

- Nincs standard megoldás!
- Tipikus szintaxis:

```
CREATE INDEX SörInd ON  
Sörök (gyártó) ;
```

```
CREATE INDEX EladásInd ON  
Felszolgál (bár, sör) ;
```

Indexek használata

- Adott v értékre az index azokhoz a sorokhoz irányít, ahol ez a v érték megjelenik a megfelelő attribútum(ok)nál.
- **Példa:** a SörInd és az EladásInd indexek segítségével megkeressük azokat a söroket, melyeket Pete gyárt és Joe árul. (köv.oldal)

Indexek használata

SELECT ár

FROM Sörök S, Felszolgál F

WHERE gyártó = 'Pete' AND

S.név = F.sör AND

bár = 'Joe bárja' ;

1. A SörInd segítségével megkapjuk azokat a söröket, melyeket Pete gyárt.
2. Aztán a EladásInd használatával a Joe bárjában felszolgált sörök árait kapjuk meg.

Adatbázisok hangolása

- Az adatbázisok hangolásánál (minél gyorsabb legyen) komoly kérdést jelent annak eldöntése, hogy milyen indexeket használjanak.
- **Mellette:** az index felgyorsíthatja a lekérdezések végrehajtását.
- **Ellene:** a módosítások lassabbak lesznek, hiszen az indexeket is módosítani kell.

Példa: hangolás

- Tegyük fel, hogy a sörös adatbázisunkban a következők történhetnek:
 1. Új tények kerülnek egy relációba (10%).
 2. Adott kocsmára és sörre keressük az ottani árat (90%).
- Ekkor az **EladásInd** a Felszolgál(kocsmá, sör) fölött nagyszerű szolgáltatást tesz, a **SörInd** a Sörök(gyártó) fölött pedig inkább a kárunkra van.

Hangolási szakértők

- Fontos kutatási feladat.
 - A kézi hangolás nagy szakértelmet kíván.
- A szakértő először egy (**query load**)-t kap kézhez:
 1. véletlen lekérdezéseket választanak a korábban végrehajtottak közül.
 2. A tervező átad egy minta.
- A szakértő létrehozza a szerinte fontos indexeket, majd megvizsgálja azok hatását.
 - Minden minta lekérdezés esetén a lekérdezés optimalizálónak használnia kell az indexeket.
 - Így meg tudja mondani, hogy javult-e összességében a lekérdezések végrehajtási ideje.

Tankönyv példája: nézettáblára

Példa: Egy olyan nézettáblát szeretnénk, mely a Film(cím, év, hossz, színes, stúdióNév, producerAzon) reláció egy részét jelképezi, pontosabban a Paramount stúdió által gyártott filmek címét és gyártási évét

```
CREATE VIEW ParamountFilm AS  
SELECT cím, év  
FROM Film  
WHERE stúdióNév = 'Paramount';
```

Tankönyv példája: triggerre

Példa: **INSTEAD OF** (helyette) típusú triggerre

```
CREATE TRIGGER ParamountBeszúrás
    INSTEAD OF INSERT ON ParamountFilm
    REFERENCING NEW ROW AS ÚjSor
    FOR EACH ROW
    INSERT INTO Film(cím, év, stúdióNév)
    VALUES (Újsor.cím, ÚjSor.év, 'Paramount');
```