

Adatbázisok 1.

Tranzakciók, nézettáblák, indexek – 2. rész

Párhuzamos folyamatok irányítása
Virtuális és materializált nézettáblák
Az adathozzáférés felgyorsítása

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.

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

Példa: nézettábla definíció

- Ihatja(alkesz, sör) nézettáblában az alkeszek mellett azon söröket tároljuk, melyeket legalább egy kocsmában felszolgálnak az általa látogatottak közül:

```
CREATE VIEW Ihatja AS
  SELECT alkesz, sör
  FROM Látogat, Felszolgál
 WHERE Látogat.kocsm =
        Felszolgál.kocsm;
```

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 alkesz = 'Sally';
```

Materializált nézettáblák

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

Példa: levelezési lista

- A következő levelezési lista [cs145-aut0708](#) valójában egy materializált nézettábla, 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 áruházlánc *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 értéke, attribútumainak értékei alapján.
- Lehet hash tábla, de az *ab* rendszerekben a legtöbb esetben kiegyensúlyozott keresési fával valósítják meg (*B-fák*).

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

Indexek deklarálása

- Nincs standard megoldás!
- Tipikus szintaxis:

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

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

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

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 söröket, melyeket Pete gyárt és Joe árul. (következő dia)

Indexek használata --- (2)

```
SELECT ár FROM Sörök, Felszolgál  
WHERE gyártó = 'Pete' AND  
       Sör.név = Felszolgál.sör AND  
       kocsmá = '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.

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

Adatbázisok hangolása

- Az adatbázisok hangolásánál 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

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

- 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(kocsm, sör) fölött nagyszerű szolgálatot 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 *lekérdezés terhelési kimutatást* (query load) kap kézhez:
 1. véletlenszerűen lekérdezéseket választanak a korábban végrehajtottak közül.
 2. A tervező átad egy mintát.

Hangolási szakértők --- (2)

- 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.