

Adatbázisok 1.

Tranzakciók, nézettáblák, indexek – 2. rész

Párhuzamos folyamatok irányítása
Virtuális és materializált nézettáblák
Az adathozzáférés felgyorsítása

Nézettáblák (*views*)

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

Nézettáblák létrehozása

- Deklaráció:

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

- Alapesetben virtualizált nézettábla jön létre.

Teák(név, gyártó)

Teázók(név, cím, engedélySzám)

Vendégek(név, cím, telefon)

Szeret(vendég, tea)

Felhasznál(teázó, tea, ár)

Látogat(vendég, teázó)

Példa: nézettábla definíció

- **Ihatja(vendég, tea)** nézettáblában a vendégek mellett azon teákat tároljuk, melyeket legalább egy teázóban felszolgálnak az általa látogatottak közül:

```
CREATE VIEW Ihatja AS
  SELECT vendég, tea
  FROM Látogat, Felszolgal
  WHERE Látogat.teázó = Felszolgal.teázó;
```

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 tea FROM Ihatja  
WHERE vendég = '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 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 áruházlánc *adattárházát (data warehouse)*, 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 (*balanced search tree*) valósítják meg (*B-fák*).

Teák(név, gyártó)

Teázók(név, cím, engedélySzám)

Vendégek(név, cím, telefon)

Szeret(vendég, tea)

Felhasználó(teázó, tea, ár)

Látogat(vendég, teázó)

Indexek deklarációja

- Nincs standard megoldás!
- Tipikus szintaxis:

```
CREATE INDEX TeaInd ON Teák(gyártó);
```

```
CREATE INDEX EladásInd ON Felhasználó(teázó, tea);
```

Indexek használata

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

Indexek használata --- (2)

```
SELECT ár FROM Teák, Felszolgal
WHERE gyártó = 'Pete' AND
      Teák.név = Felszolgal.tea AND
      teázó = 'Joe teázója';
```

1. A Tealnd segítségével megkapjuk azokat a teákat, melyeket Pete gyárt.
2. Aztán a EladásInd használatával a Joe teázójában felszolgált teák árait kapjuk meg.

Teák(név, gyártó)

Teázók(név, cím, engedélySzám)

Vendégek(név, cím, telefon)

Szeret(vendég, tea)

Felszolgal(teázó, tea, ár)

Látogat(vendég, teázó)

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 (*tuning*)

- Tegyük fel, hogy a teás adatbázisunkban a következők történhetnek:
 1. Új tények kerülnek egy relációba (10%).
 2. Adott teázóra és teára keressük az ottani árat (90%).
- Ekkor az **EladásInd** a Felszolgál(teázó, tea) fölött nagyszerű szolgálatot tesz, a **TeaInd** a Teák(gyártó) fölött pedig inkább a kárunkra van.

Teák(név, gyártó)

Teázók(név, cím, engedélySzám)

Vendégek(név, cím, telefon)

Szeret(vendég, tea)

Felszolgál(teázó, tea, ár)

Látogat(vendég, teázó)

Hangolási szakértők (*tuning advisors*)

- 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** (*running time*).