

Tietokantojen hakemistorakenteet

- Hakemistorakenteiden (**indeksien**) tarkoituksena on nopeuttaa tietojen hakua tietokannasta.
- Hakemisto voi olla 'ylimääräinen' **oheishakemisto (secondary index)**, esimerkiksi kasarakenteen päälle rakennettu rakenne, joka tarjoaa vaihtoehoisen saantipolun, joidenkin kyselyjen toteutukseen
 - oheishakemistoja voi tiedostoon liittyä useita eri perustein muodostettuja
- Hakemisto voi olla myös välttämätön osa tiedostorakennetta (**primary index, clustered index**). Tällöin tiedoston tietueet järjestellään hakemiston tarpeiden mukaisesti.

Hakemistotyypeistä

- Hakemisto
 - koostuu **hakemistomerkinnoistä (index entry)**
 - perustuu johonkin **muodostusperustaan (indexing field)**, eli yhteen tai useampaan tietueen kenttään
 - on relaatiotietokantojen yhteydessä yleensä kentän **koko arvoon** perustuva
 - tietueesta on samassa hakemistossa enintään yksi hakemistomerkinä
 - vrt. esim. tekstitietokannoissa samaan hakemistoon voi tulla useita merkintöjä saman kentän (teksti) perusteella (jokainen kentässä oleva sana aiheuttaa merkinnän)

Hakemistotyypeistä

Kokonaisiin arvoihin perustuva indeksi (kenttä TEXT):

TEXT:

DATA

avainsanaindeksi (samasta kentästä useita merkintöjä)

Hakemistotyypeistä


- Hakemisto voi olla
 - tiheä (dense)**
 - tiheässä hakemistossa on hakemistomerkinä jokaista tiedoston tietuetta kohti (taulun riviä kohti)
 - harva (sparse)**
 - harvassa hakemistossa on yksi hakemistomerkinä jokaista tietyllä periaatteella määräytyvää tietuejoukkoa kohti

Hakemistotyypeistä


- Hakemistomerkinä sisältää vähintään
 - hakemistoavaimen (indexing key)**
 - muodostusperustan määrittelemänä tietueesta tai tietuejoukosta tuotettu tunnus – yleensä suoraan kentän arvo
 - yhden tietuetunnisteen** tai
 - joukon tietuetunnisteita**
 - (mikäli sama avain esiintyy useassa tietueessa)

Hakemiston toteutuksesta


- Teknisesti **hakemistokin on tiedosto**
 - muodostuu sivuista (hakemistosivu)
 - hakemistomerkinat ovat tietueita
 - tarvitsee käsittelyä varten puskureita
 - koska useat tietokantahaut saattavat edellyttää hakemiston käyttöä pyrkivät tkhj:ien puskurienhallintarutiinit suosimaan hakemistosivujen säilymistä puskureissa

 Hakemiston toteutuksesta


- Hakemisto voitaisiin toteuttaa aiemmin käsiteltyjen tiedostorakenteiden avulla
 - kasa, järjestetty peräkkäistiedosto, hajautusrakenne
 - näistä käytännössä käytössä on vain hajautusrakenne ns. hash index rakenteena.
- Hakemistoja varten on kehitetty myös erityisiä hakemistokäyttöön tarkoitettuja rakenteita (esim. B+-puu, tarkastellaan myöhemmin)

 Hakemiston käyttö


- Haku hakemistoa käyttäen on kaksivaiheista
 - ensin etsitään hakemistomerkintä hakemistosivuilta ja
 - hakemistomerkinnän perusteella haetaan tietueen (tietueet) sisältävä sivu(t)
- Tietueen hakua varten tarvitaan siis vähintään kaksi levyhakua (elleivät sivut ole puskurissa)
- Hakemistotietuetta voidaan joutua etsimään usealta hakemistosivulta.
 - Koska hakemistomerkinnät ovat yleensä lyhyempiä kuin varsinaiset tietueet, niitä mahtuu sivulle useampia ja sivuja on vähemmän
 - **Seuraus: hakemiston kautta on nopeampi etsiä tietuetta**

 Hakemiston käyttö


- Ns. **lihava hakemisto** (fat index) sisältää varsinaisen hakukriteeritiedon lisäksi toistettua tietokannan dataa hakemistomerkinnässä
 - esimerkiksi opiskelijatietojen hakua varten riittäisi tehdä hakemisto opiskelijanumeron perusteella, mutta koska opiskelijanumerolla haettaessa lähes aina kysytään opiskelijan nimeä otetaan nimikin mukaan hakemiston hakemistoavaimen
 - jos haku kohdistuu pelkästään hakemistomerkinnästä löytyvään tietoon ei varsinaista tietuetta tarvitse hakea lainkaan.
- Esimerkiksi Oracle tarjoaa yhtenä vaihtoehtona taululle 'index only' -toteutusta. Tässä ratkaisussa ei ole lainkaan datatietueita vaan kaikki data on hakemistomerkinnöissä (tulee kyseeseen B+-puu toteutuksen yhteydessä).

 Hakemiston käyttö


- Muutokset tiedostossa saattavat edellyttää muutoksia hakemistoon
 - Tietueen lisäys tiedostoon edellyttää hakemistomerkinnän lisäämistä (tai merkintöjen muuttamista) kaikkiin kyseiseen tiedostoon liitettyihin tiheisiin hakemistoihin
 - Tietueen poisto tiedostosta edellyttää tietueeseen liittyvien hakemistomerkintöjen poistamista (tai mitätöintiä) ainakin tiheissä hakemistoissa
 - Hakemiston muodostusperustana olevan kentän muuttaminen edellyttää hakemistomerkinnänkin muuttamista (yleensä edellisen poistoa ja uuden vientiä hakemistoon)

 Hakemiston käyttö


- Jos hakemistomerkinnässä käytettävät tietuetunnisteet voivat muuttua saattaa tietueen todellinen poisto sivulta (niin että seuraavien tietueiden järjestysnumerot muuttuvat) edellyttää useiden hakemistomerkintöjen muuttamista eri hakemistosivuilla
 - siksi tietuetunnisteet eivät yleensä muutu vaan kerran käyttöönotettu tunnus säilyy uudelleenorganisointiin asti (poistot poistoleimalla)

 Hakemiston käyttö


- Tarkastellaan esimerkkinä työntekijä-taulua:
 - työntekijännumero (avain, 10 merkkiä)
 - nimi (enintään 40, keskimäärin 20)
 - osoite
 - palkka
 - osasto_nro (4 merkkiä)
 - jne, yht. keskimäärin 300 tavua.
 - Taulussa 8000 riviä.
 - Tietoja haetaan lähinnä työntekijännumerolla, nimellä ja osastonumerolla

 Hakemiston käyttö

- Olkoon osoitteen pituus 6 tavua ja hallintatietoa olisi 4 tavua yhtä hakemistomerkintää kohti tällöin merkintöjen koot olisivat
 - a) Työntekijännumero – 20 tavua
 - b) Työntekijän nimi - keskimäärin 30 tavua
 - c) Osastonnumero – tietuepituus riippuu taulukoitujen osoitteiden määrästä $4+4+n*6$, jos $n=200$, niin 1208 tavua
 - Jos hakemisto olisi järjestetty peräkkäistiedosto, täyttösuhde 70% niin hakemistosivuille (4KB) menisi
 - a) 140 merkintää -> yhteensä 58 lohkoa
 - b) 93 merkintää -> yhteensä 87 lohkoa
 - c) 200 merkintää -> yhteensä $\lceil 40/3 \rceil = 14$ lohkoa

 Hakemiston käyttö

- Kaikki hakemistot ovat niin pieniä, että ne kannattaa lukea peräkkäislukuna tarvittaessa
- a) haku aika työntekijänumerolla olisi keskimäärin
 $10\text{ms} + (1/2) * 10 * 58 / 50 \text{ ms} + 10 \text{ ms} = 25.8 \text{ ms}$
(kohdistus+pyörähdysviive) + keskimäärin siirto + datasiivu)
- b) Haku aika nimellä veisi
 $10\text{ms} + (1/2) * 10 * 93 / 50 \text{ ms} + 10 \text{ ms} = 29.3 \text{ ms}$
- Näissä on oletettu, että haku tuottaa yhden osuman, jolloin päästään noin kolmannekseen kasan keskimääräisestä hakuajasta
- (levy sama 10 ms hajasaantiajan levy kuin aiemmin)

 Hakemiston käyttö

- Haettaessa osastonumerolla osumia tulee useampia. Oletetaan että osastoja on 40, jolloin yhdellä osastolla on keskimäärin 200 työntekijää
- Osaston työntekijöiden haku indeksii käyttäen veisi aikaa:
 $10 \text{ ms} + (1/2) * 10\text{ms} * 14 / 50 + 200 * (10 + 0.2) \text{ ms} = 10\text{ms} + 1.4\text{ms} + 2040\text{ms} = 2051.4 \text{ ms} = \text{noin } 2\text{s}$
- Indeksistä **ei ole hyötyä** sillä aiemmin laskettiin koko kasan lukemiseen menevän vain **n 180ms**.