## Tietokantojen hakemistorakenteet

- Hakemistorakenteiden (indeksien) tarkoituksena on nopeuttaa tietojen hakua tietokannasta.
- Hakemisto voi olla 'ylimääräinen' oheishakemisto (secondary index), esimerkiksi kasarakenteen päälle rakennettu rakenne, joka tarjoaa vaihtoehtoisen 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.



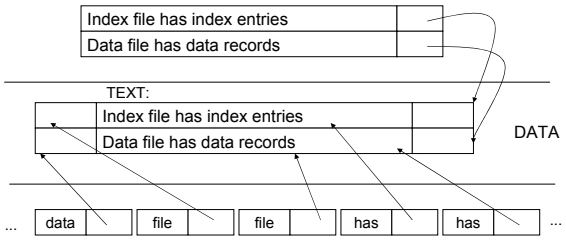
## Hakemistotyypeistä

- Hakemisto
  - koostuu hakemistomerkinnöistä (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 hakemistomerkintä
    - 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):




avainsana indeksi (samasta kentästä useita indeksirivejä)



## Hakemistotyypeistä

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



## Hakemistotyypeistä

- Hakemistomerkintä sisältää
  - hakemistoavaimen (indexing key)
    - muodostusperustan määrittelemänä tietueesta tai tietuejoukosta tuotettu tunnus – yleensä suoraan kentän arvo
  - tietueen osoitteen, joka voi olla joko
    - sivuosoite tai
      - *tietueen siirtely sivun sisällä helppoa*
    - riviosoite (sivuosoite + rivin (tietueen) järjestysnumero)
      - *tietueen siirtely edellyttää tietueeseen viittavien hakemistomerkintöjen muuttamista, tietue löytyy nopeammin*



## Hakemiston toteutuksesta

- Teknisesti hakemistokin on tiedosto
  - muodostuu sivuista (hakemistosivu)
    - hakemistomerkinnät 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
- 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 sisältävä sivu
- 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 / indeksikentässä
  - esimerkiksi hakua varten riittäisi tehdä hakemisto opiskelijanumeron perusteella, mutta koska opiskelijanumerolla haettaessa lähes aina kysytään opiskelijan nimeä otetaan nimikin mukaan hakemiston indeksointiavaimeen
  - 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 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)



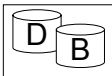
## Hakemiston käyttö

- Jos hakemistomerkinnässä käytetään riviosoitetta saattaa tietueen todellinen poisto sivulta (niin että seuraavien tietueiden järjestysnumerot muuttuvat) edellyttää useiden hakemistomerkintöjen muuttamista eri hakemistosivuilla



## Hakemiston käyttö

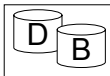
- Tarkastellaan esimerkkinä työntekijä-taulua:
  - työntekijänumero (avain, 10 merkkiä)
  - nimi (max 40, avg 20)
  - osoite
  - palkka
  - osastonro (4 merkkiä)
  - jne, yht keskimäärin 300 tavua.
  - Taulusssa 8000 riviä.
  - Tietoja haetaan lähinnä työntekija



## Hakemiston käyttö

- Olkoon osoitteen pituus 6 tavua ja hallintatietoa olisi 4 tavua yhtä hakemistomerkintää kohti tällöin merkintöjen koot olisivat
  - a) Työntekijänumero – 20 tavua
  - b) Työntekijän nimi - keskimäärin 30 tavua
  - c) Osastonumero – 14 tavua
  - Jos hakemisto olisi järjestetty peräkkäistiedosto, täyttösuhde 70% niin hakemistosivuille menisi
  - a) 140 merkintää -> yhteensä 58 lohkoa
  - b) 93 merkintää -> yhteensä 87 lohkoa
  - c) 200 merkintää -> yhteensä 40 lohkoa





## Hakemiston käyttö

- Kaikki hakemistot ovat niin pieniä, että ne kannattaa lukea peräkkäislukuna tarvittaessa
- a) hakuaika työntekijänumerolla olisi keskimäärin
  10ms+ (1/2)*10*58/50 ms +10 ms = alle 25.8 ms
- b) Hakuaika nimellä veisi
  10ms + (1/2)* 10*93/50 ms + 10 ms = 29.3 ms
- Näissä on oletettu, että haku tuottaa yhden osuman, jolloin päästään hieman 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 indeksiä käyttäen veisi aikaa:
  10 ms + (1/2)*10ms*40/50 + 200*10 ms =2014ms
- Indeksistä ei ole hyötyä sillä aiemmin laskettiin koko kasan lukemiseen menevän vain n 170ms.