

Harvat hakemistot

- Harvassa hakemistossa on ei ole hakemistomerkintöjä jokaista tietuetta kohden vaan yksi merkintä jotain isompaa kokonaisuutta esimerkiksi sivua tai sivujoukkoa (esim. saman uran sivut) kohti
- Harvan hakemiston käyttö edellyttää, että tietueet on järjestetty

1

Harvat hakemistot

hakemistossa alarajat sivulle kuuluville avaimille ja osoitteet sivuille

pienin mahdollinen

kaikki tietueet, joissa $70 \leq \text{avain} < 110$ kuuluvat sivulle 2

perustietueet järjestettynä peräkkäisrakenteena

2

Harvat hakemistot

hakemistossa alarajat sivulle kuuluville avaimille
(Voidaan toteuttaa myös ilman osoitteita, hakemistomerkinnän indeksi vastaa sivunumeroa)

pienin mahdollinen

kaikki tietueet, joissa $70 \leq \text{avain} < 110$ kuuluvat sivulle 2

perustietueet järjestettynä peräkkäisrakenteena

3

Harvat hakemistot

- perustietueet järjestettynä peräkkäistiedostona
- hakemistoon alarajat (tai ylärajat) sivulle kuuluvien tietueiden avaimille (+osoitteet sivuille)
- kaksi peräkkäistä hakemistoavainta määrää sivulle i kuuluvat avaimet
- $h(i) \leq \text{sivulle } i \text{ kuuluva avain} < h(i+1)$
- hakemisto voi olla monitasoinen
 - jos alin hakemistotaso tulee liian isoksi, tehdään sille samoin harva hakemisto

4

Harvat hakemistot

ylempi hakemistotaso

alin hakemistotaso


hakemistotasojia voisi olla useitakin, mutta yleensä 2 tai 3 riittää siihen, että ylimmän tason hakemisto on niin pieni, että se voidaan pitää jatkuvasti keskusmuistissa.

5

Harvat hakemistot


- Haku:
 - etsi hakemistosivulta suurin hakemistoavain, joka on pienempi tai yhtä suuri kuin etsittävän avain,
 - jos ollaan hakemistosivulla, siirry löydetyn alkion indeksin osoittamalle sivulle seuraavalla tasolla (jos hakemistomerkintöihin liittyy sivuosoite käytetään sitä indeksin tilalla)
 - tässä joudutaan suorittamaan levyhaku (ellei sivu ole puskurissa)
 - jos ollaan datasisivulla, etsi tietue sivulta

6

 Harvat hakemistot


- Jos datasiivulle mahtuu useita tietueita tulee harva hakemisto tiheää pienemmäksi, joten haku on nopeampaa
- Esimerkki
 - työntekijätietueen keskipituus oli 300, joten yhdelle datasiivulle voitaisiin sijoittaa vaikkapa 10 tietuetta (jätetty hieman tilaa lisäyksille)
 - Tiedosto mahtuu 800 sivulle
 - Kunkin sivun pienin avain viedään hakemistoon, jolloin hakemistosivulle menee keskimäärin 400 avainta (tässä muunnelmassa ei käytetä osoitteita)
 - Tarvitaan 2 hakemistosivua alimmalla tasolla,
 - Koko hakemisto (yht. 8K) voidaan pitää keskusmuistissa, joten tietueen hakuun työntekijänumeron perusteella tarvitaan noin 1 levyhaku

7

 Harvat hakemistot


- Etuja
 - pieni hakemisto - tehokas haku
 - jos tietueita on N ja hakemistosivulle mahtuu X merkintää, tarvitaan hakemistoon $\text{roof}(\log_x N)$ tasoa
 - tukee myös
 - arvovälihakuja
 - hakuja avaimen alkuosalla
 - (osoite like 'Helsingin%')

8

 Harvat hakemistot


- Ongelmia
 - tiedostolla voi olla **vain yksi harva hakemisto** sillä perustietueet voidaan järjestää vain yhteen järjestykseen
 - lisäykset voivat aiheuttaa ongelmia
 - jos lisäykset hoidetaan sivukohtaisilla ketjutetuilla ylivuotosivuilla, voi ketjusta tulla pitkä, jos lisäykset kasautuvat
 - usein jätetään sivuille tilaa lisäyksille – täyttösuhde huononee eikä tämä välttämättä hidasta kovin paljon rakenteen degeneroitumista
 - tietueiden lisääminen satunnaisessa järjestyksessä on erityisen ongelmallista

9

 Harvat hakemistot - ISAM


- ISAM (indexed sequential access method)
- indeksoitu peräkkäisrakenne
- harva **monitasoinen staattinen** hakemisto
 - staattinen: hakemisto pysyy luontinsa jälkeen muuttumattomana kunnes tiedosto organisoidaan uudelleen ja sille luodaan uusi hakemisto
 - tiedosto jakautuu **perussivuihin** ja **ylivuotosivuihin**
 - ylivuotosivuilla viedään lisäykset, jotka eivät mahdu perussivuille, ylivuotosivut ketjutetaan

10

 Harvat hakemistot - ISAM

- Tiedosto **luodaan järjestetyn aineiston pohjalta**
 - Tietueita lisätään tiedoston loppuun (järjestyksessä)
 - Sivuille jätetään tilaa lisäyksille, esim. 30 % - 50%.
 - Hakemistomerkinnässä **avainarvo** ja **sivunosoite**
 - Viedään merkintä hakemistosivulle aina kun aloitetaan uusi datasiivu
 - ensimmäisen hakemistosivun kohdalla viedään hakemisto-avaimeksi **low-value** = pienin mahdollinen arvo, muille sivuille sivun ensimmäisen tietueen avain
 - Hakemistosivut voidaan laittaa täyteen, sillä niitä ei päivitetä.

11

 Harvat hakemistot - ISAM

- Rakenne sopii hyvin tiedostoille, joihin tulee vain vähän lisäyksiä. Jos lisäyksiä tulee runsaasti tarvitaan ajoittaisia uudelleenorganisointeja.
- Ylivuotoketjut hidastavat läpilukuaikaa samoin kuin järjestetyssä peräkkäisrakenteessa.
- Koska hakemistosivut eivät muutu, ei niitä myöskään tarvitse lukea rinnakkaisessa käsittelyssä
 - prosessin ei tarvitse jonottaa hakemistosivua

12

D B Harvat hakemistot - ISAM

- Poistot
 - kun ylivuotosivu tyhjenee poiston seurauksena sivu irrotetaan ketjusta ja voidaan ottaa uudelleen käyttöön
 - kun datasisivu tyhjenee poiston seurauksena se jätetään tyhjäksi. Sivun tarjoaa tällöin lisäyspaikan, jos sen avainalueelle sattuisi tulemaan lisäyksiä.

13

D B Harvat hakemistot - ISAM

- Alkuperä:
 - IBM:n ISAM rakenteessa oli alunperin kaksi hakemistotasoa alimpana urahakemisto (uran suurin avain), ja seuraavalla tasolla sylinterihakemisto (sylinterin suurin avain)
- Muunnelmia
 - ISAM rakennetta voidaan käyttää tiheän hakemiston toteutusrakenteena, siten että 'datatietueina' eivät olekaan 'oikeat datatietueet' vaan tiheän hakemiston hakemistomerkinnot.

14

D B Harvat hakemistot - ISAM

- ISAM on keskimäärin hyvin tehokas rakenne hakemistoavaimen avulla tehtävissä hakuoperaatioissa (yhtäsuuruus, alkuosa, arvoväli)
- Rakenne ei kykene takaamaan kasvavalle tiedostolle kovin pientä ylärajaa levyhakujen määrälle pahimmassa tapauksessa, jolloin levyhakuja olisi luokkaa:

lisäysten määrä/sivukoko

15

D B Dynaamiset hakemistorakenteet B+ -puu

- B+ -puut ovat laajalti tietokantojen yhteydessä käytetty tiedostorakenne
- Rakenne on oikeastaan ISAM rakenteen dynaaminen muunnelma
 - Rakenteen alimmalla tasolla ovat datatietueet järjestettynä rakenteena kuten ISAM:ssa,
 - Datasivut eivät kuitenkaan ole välttämättä fyysisesti peräkkäisiä vaan ne on usein kytketty kaksisuuntaiseksi linkitettyksi listaksi. Erillisiä ylivuotosivuja ei ole, vaan listarakennetta ylläpidetään dynaamisesti.
 - Datasivujen 'päällä' on monitasoinen harva hakemisto, joka ISAM:ista poiketen on dynaaminen

16

D B Dynaamiset hakemistorakenteet B+ -puu

hakemisto voi laajeta ja kutistua

harva hakemisto

Datisivut (sequence set)

datisivujen joukko voi laajeta ja kutistua

17

D B Dynaamiset hakemistorakenteet B+ -puu

- B+ -puun hakemisto on rakenteeltaan ISAM:in kaltainen, mutta
 - Datasivujen täyttöaste 50-100% samoin hakemistosivujen ylittä sivua lukuun ottamatta.
 - hakemiston alaisuuteen kuuluvan ensimmäisen sivun pienin avainarvo jää pois (osoitteet ja erottimet)

18

D B Dynaamiset hakemistorakenteet B+ -puu

- hakemistosivut **eivät ole** staattisia, niihin voi tulla uusia hakemistomerkintöjä, merkintöjä voidaan poistaa, sivuja voidaan jakaa ja yhdistää.
- B+ -puun tasapainovaatimuksena on, että jokainen **hakupolku on yhtä pitkä**, eli jokaisen tietueen hakemiseen tarvitaan yhtä monta levyhakua (elleivät sivut ole puskureissa)
- lisäys- ja poisto-operaatiot säilyttävät hakurakenteen tasapainon

19

D B Dynaamiset hakemistorakenteet B+ -puu

- Tarkastellaan **kertaluokan (order) d** B+ -puuta
- Tällaisen puun jokaisessa **hakemistosolmussa** paitsi juuressa (ylin taso) on $d \leq m \leq 2d$ erotinta ja $m+1$ osoitetta
- hakemistosivulla ovat siis osoitteet p_0, \dots, p_m ja etrottimet K_1, \dots, K_m
- Kuten ISAM:ssa tietue kuuluu osoitteessa p_i ($i=1, \dots, m$) olevan sivun alaisuuteen, jos sen avaimelle pätee $K_i \leq \text{avain} < K_{i+1}$
- Tietue kuuluu osoitteessa p_0 olevan sivun alaisuuteen, jos sen **avain** $< K_1$

20

D B Dynaamiset hakemistorakenteet B+ -puu

- Haku (oletetaan, että avaimet yksikäsitteisiä)

```
function search(node, key) {
  if (node is leaf) return node;
  else
    if (key < K[1]) return search(p[0], key);
    else
      if (key >= K[m]) return search(p[m], key);
      else {
        find i such that K[i] <= key < K[i+1];
        return search(p[i], key);
      }
}
```

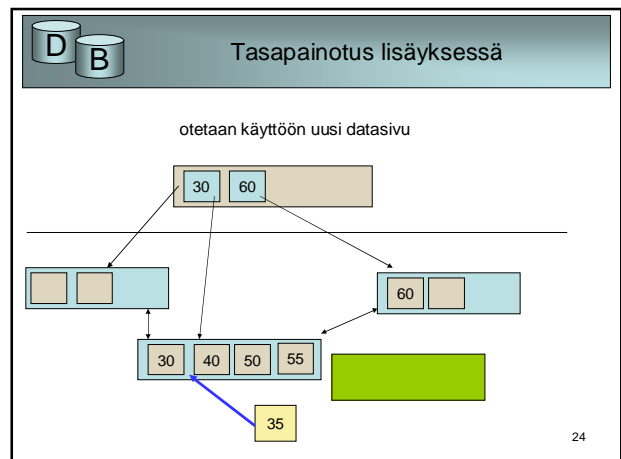
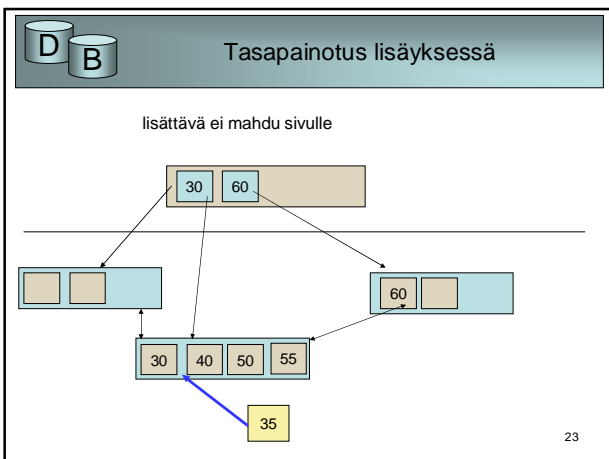
Palauttaa datasivun osoitteen

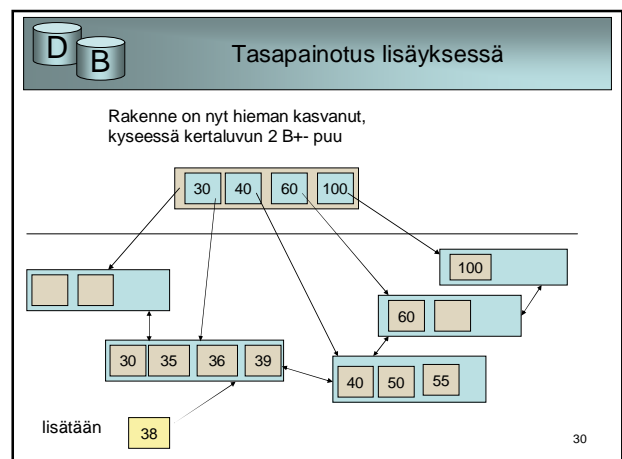
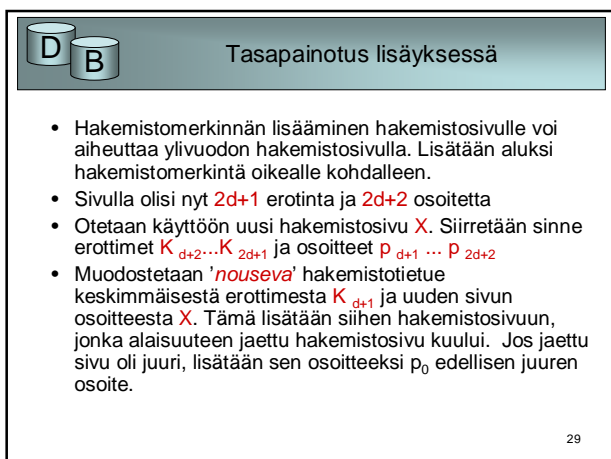
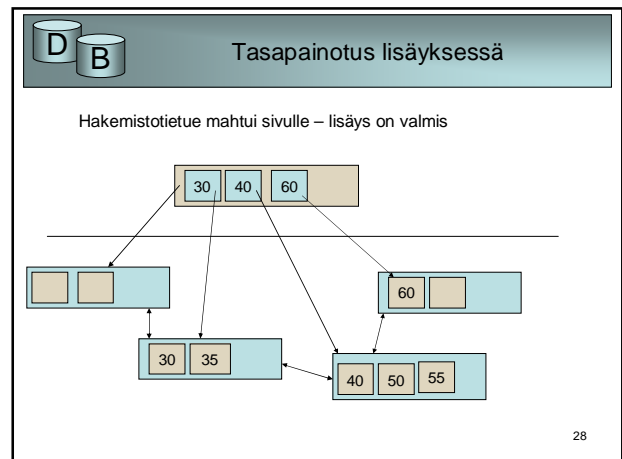
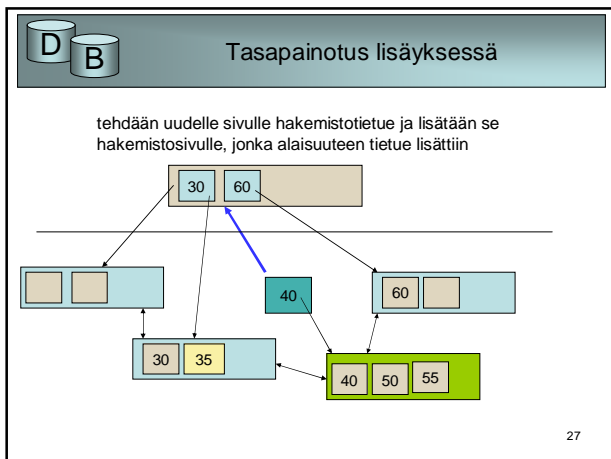
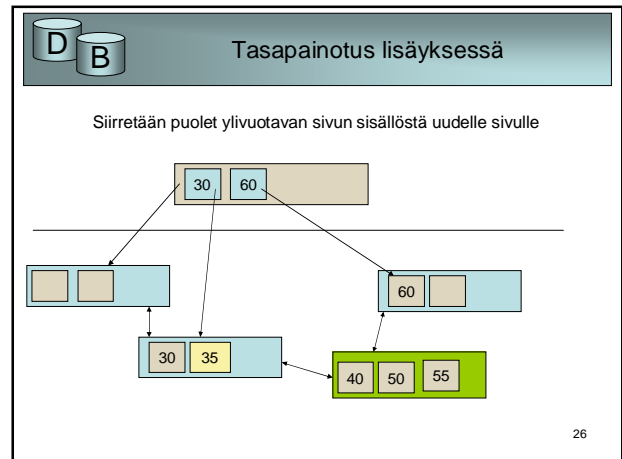
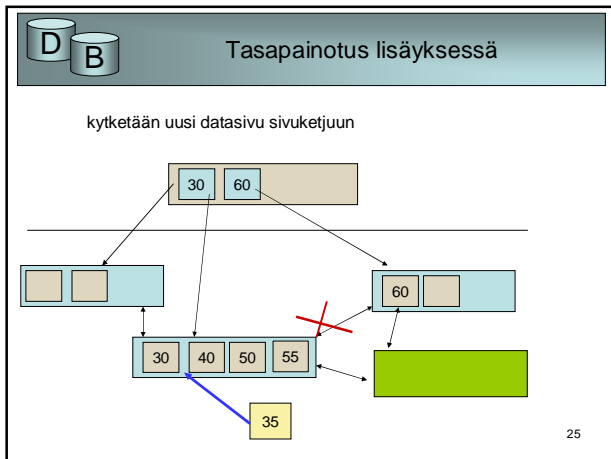
21

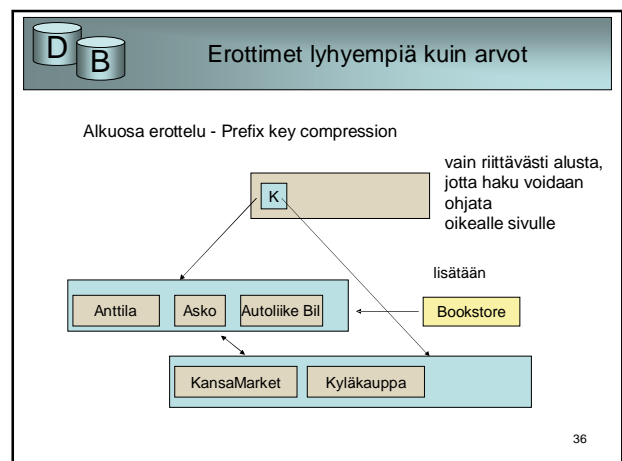
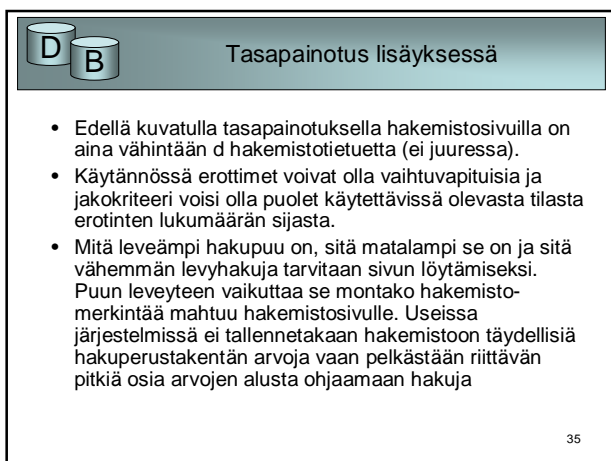
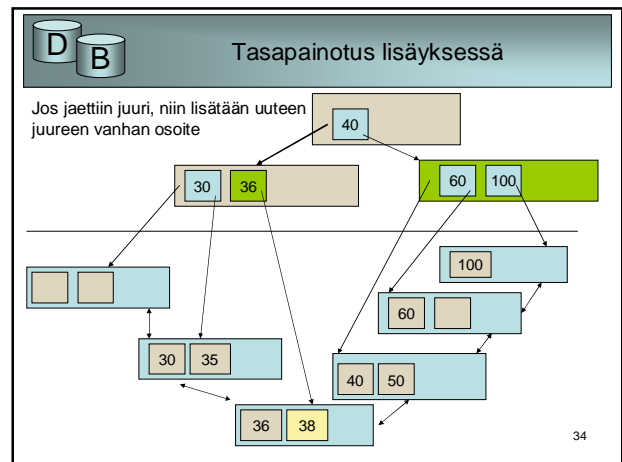
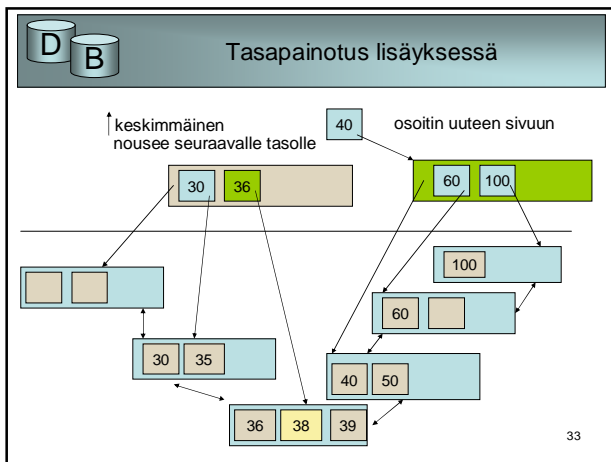
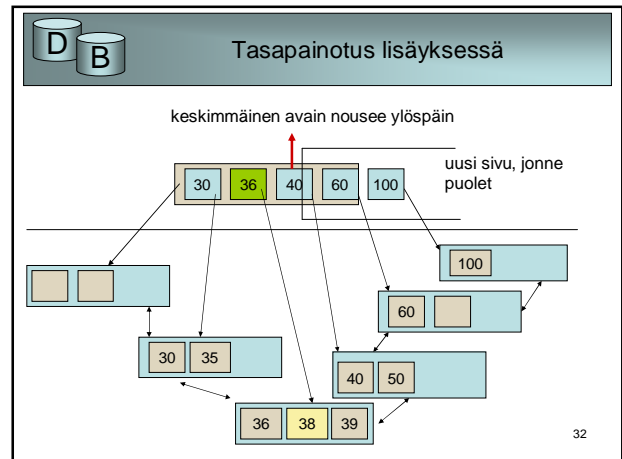
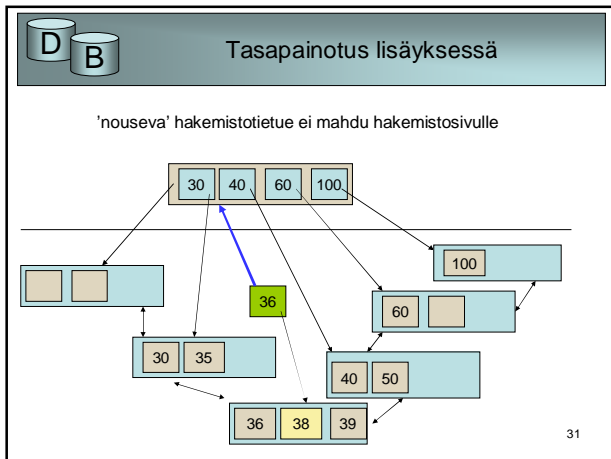
D B Dynaamiset hakemistorakenteet B+ -puu

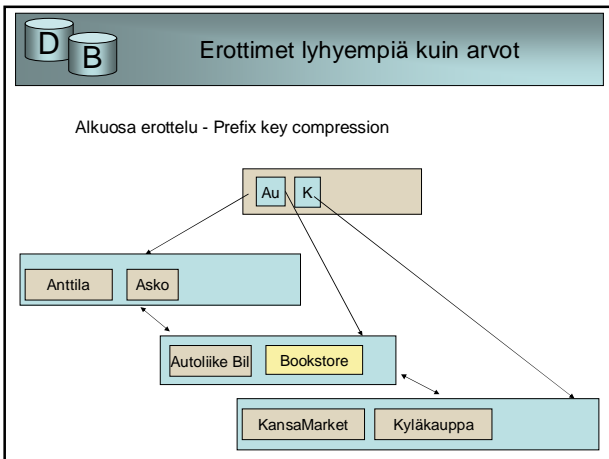
- Lisäykset:
 - Lisättäessä tietuetta B+ -puuhun sille haetaan sijoituspaikka edellä kuvatulla hakuoperaatiolla
 - Jos sivulla on tilaa, tietue lisätään järjestyksessä oikealle paikalleen sivulla
 - Jos tietue ei mahdu sivulle se sijoitetaan loogisesti oikealle paikalleen ja jaetaan ylivuotanut sivu:
 - Otetaan käyttöön uusi sivu ja kytetään se linkitettyyn listaan
 - Siirretään puolet alkuperäisen sivun tietueista uudelle sivulle
 - Muodostetaan hakemistomerkintä ($K_{\text{new}}, P_{\text{new}}$) uuden sivun ensimmäisen tietueen avaimesta ja sivun osoitteesta ja lisätään tämä siihen hakemistotietueeseen, jonka alaisuuteen jaettu sivu kuului

22





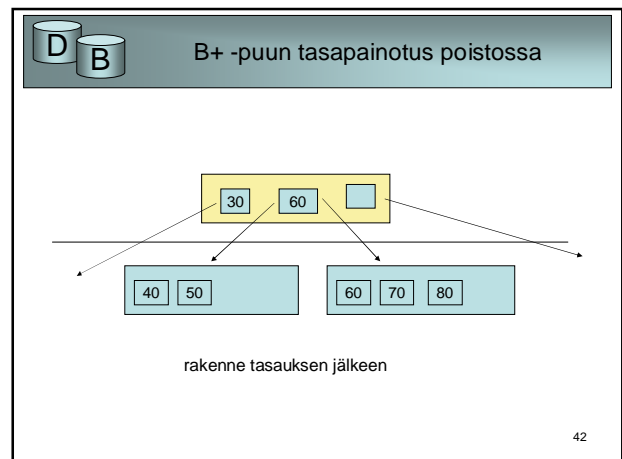
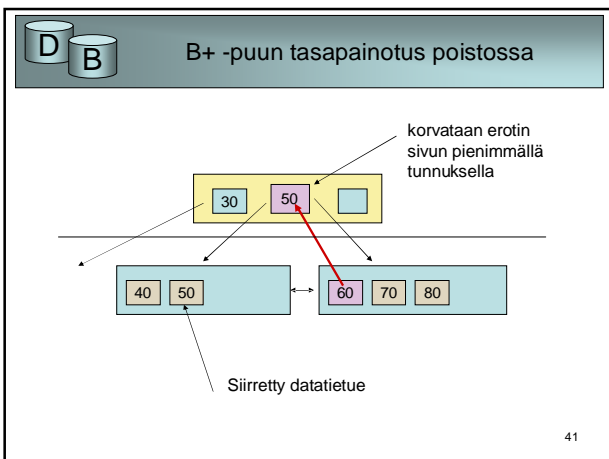
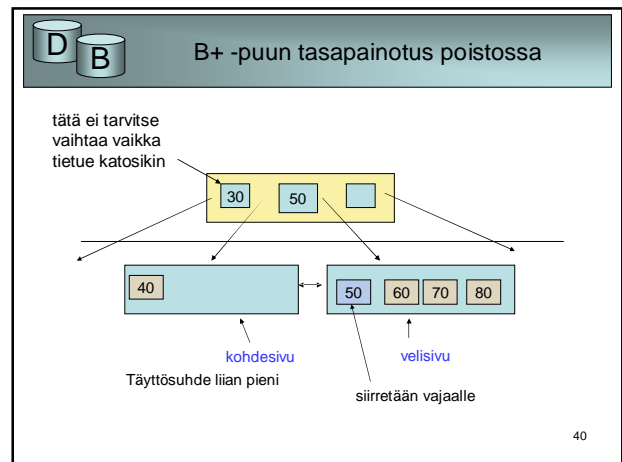
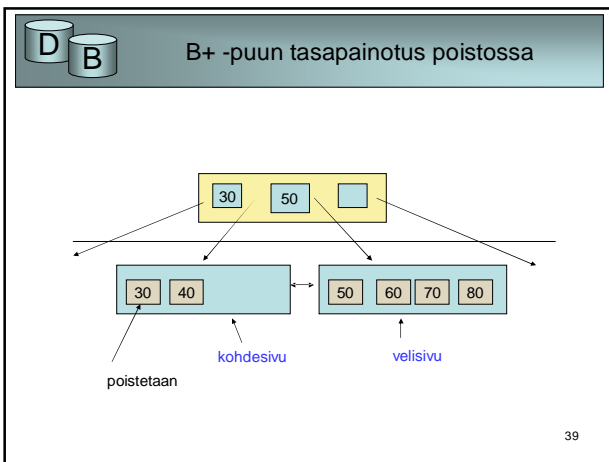




D B B+ -puun tasapainotus poistossa

- Poistot
 - Alkuperäisen B+ -puun idean mukaisesti tasapainotusta tehdään myös poistossa
 - Jos datasisivun täyttösuhde laskee alle puoleen ja sisivun ja sen velisisivun (sibling, saman isäsivun alla oleva vierussivu) yhteenlaskettu tietomäärä ylittää ylittää sivukoon, järjestetään sivuparin tiedut uudelleen siirtämällä täydemmältä sivulta tietue vajaan sivulle.

38



B+ -puun tasapainotus poistossa

- Jos poiston kohteena olevan sivun ja sen velisivun yhteenlaskettu tietomäärä jää alle sivun kapasiteetin yhdistetään sivut ja poistetaan niiden välinen erotin hakemistosta.

43

B+ -puun tasapainotus poistossa

siirretään kaikki velisivun tiedut kohdesivulle, vapautetaan tyhjentyneet sivut, poistetaan erotin hakemistosivulta

44

B+ -puun tasapainotus poistossa

45

B+ -puun tasapainotus poistossa

- Hakemistomerkin poiston takia hakemistosolmu voi jäädä vajaatäyttöiseksi
- Sivun täyttösuhdetta korjataan samalla periaatteella kuin datasisuvujenkin eli siirtämällä velisolvusta täydennystietue, **siirto tehdään isäsolmun kautta vyöryttämällä**
- Vyörytys:** Olkoon kyseessä oikeanpuoleinen veli eli velisolvussa on isompia avaimia. Vyörytys tapahtuu tällöin seuraavasti
 - Lisätään isäsolmusta kopioitu kohdesolmun ja velisolmun välinen erotin kohdesolmun erotintaulukon loppuun
 - Lisätään velisolmun ensimmäinen osoitin p_0 kohdesolmun osoitintaulukon loppuun
 - Korvataan isäsolmussa ollut kohdesolmun ja velisolmun välinen erotin velisolmun erotintaulukon pienimmällä avaimella
 - Poistetaan velisolmun avain- ja osoitintaulukkojen ensimmäinen erotin ja osoitin
 - Sivuja yhdistettäessä vyörytetään kaikki velisolmun tiedut kohdesolmuun ja poistetaan isäsolmusta erotin sekä osoitin velisolmuun
- Puun korkeus alenee, jos poistettavana on juurisolmun ainoa erotin.

46


Vyörytys

47

B+ -puun tasapainotus poistossa


- Kaikki toteutukset eivät käytä tasapainotusta poistojen yhteydessä, vaan antavat täyttöasteen laskea.
- B+ puun hakemistosivujen keskimääräiseksi täyttöasteeksi muodostuu lisäys- ja poistotasapainotusta käytettäessä noin 67%

48

B+ -puun ominaisuuksia

- Olkoon B+ -puun korkeus h
 - haettaessa indeksointiavaimen perusteella joudutaan tutkimaan h sivua ($\leq h$ levyhakua)
 - lisättäessä joudutaan lukemaan h sivua ja kirjoittamaan vähintään 1 ja enintään $h+2$ sivua
 - poistettaessa joudutaan lukemaan vähintään h ja enintään $2h-1$ ja kirjoittamaan vähintään 1 ja enintään $2h-1$ sivua
 - Indeksointiavaimen arvon muuttaminen täytyy B+ -puun yhteydessä hoitaa poisto ja lisäys operaatioina.

49

B+ -puut tietokantatoteutuksessa

- Useimmat tietokannan hallintajärjestelmät mahdollistavat B+ -puun käytön joko
 - taulun toteutusrakenteena
 - datasiivuilla on taulun rivejä vastaavia tietueita
 - ratkaisun huono puoli on se, että rivit voivat vaihtaa sivua, mikä tekee oheishakemistojen käytön työlääksi,
 - kaikki osoitteet sivua vaihtaneeseen tietueeseen on vaihdon yhteydessä päivitettävä
 - Oraclessa tämä on ratkaistu siten, että B+ -puu -rakenteisen tiedoston oheishakemisto ei käytäkään osoittamiseen tietueen osoitetta vaan tietueen pääavainta (haku hidastuu)
 - tiheän hakemiston toteutusrakenteena
 - datasiivuilla on tiheän hakemiston hakemistomerkintöjä
 - tämä on yleisin B+ -puun käyttötilanne.
 - taulun rivit ovat tällöin yleensä kasarakenteena

50