3. Tietokannan hakemistorakenteet

Tiedoston tietueiden haku voi perustua johonkin monesta **saantipolusta** (access path):
- perustiedoston tiedostorakenne
- hakemistot, joita voidaan tehdä käsittelytarpeiden mukaan perusavaimelle tai muille attribuuteille (tai attribuuttiyhdelmille)

Tiedoston **hakemisto (index)** on 'ylimääräinen' tietorakenne¨, jonka tarkoituksena on nopeuttaa tiedon hakua tiedostosta.
(Vrt. kasan ja järjestetynkin peräkkäisrakenteen hitaus.)

- ylimääräisyys:
  - perustiedosto esim. kasa (riittää periaatteessa, mutta on tehoton)
  - hakemisto esimerkiksi kaikista avaimista: järjestetty tiedosto, joka tarjoaa avainten mukaisen järjestyksen
  - hakemisto luodaan jo valmiina olevalle perustiedostolle

Hakemistorakenteet - yleistä

Hakemiston yleinen malli:

- tiedosto F, jonka tietueet ovat sivuilla P1, P2, ... , PN
- indeksointikenttä (indexing field) eli **hakemistoavain** (indexing key): yhden tai useamman kentän yhdistelmä, jonka arvoihin hakemisto perustuu
- hakemisto muodostuu tietueista (v, p), missä v on indeksointikentän arvo ja p on osoitin tiedostoon F (p voi olla jakso-osoitin tai tietueosoitin)

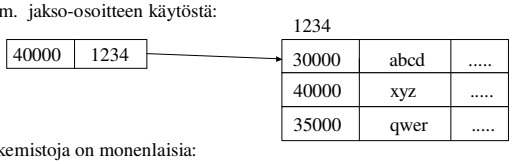
Esim. kirjan asiasanahakemisto:

binary search	138, 591
buffering	127-128

(analogia ontuva: kirjassa ei tietueita; sana ('avain') toistuu, osoitteet monimuotoisia …)

Hakemistorakenteet, yleistä - 2

Esim. jakso-osoitteen käytöstä:



Hakemistoja on monenlaisia:

- **yksitasoinen, monitasoinen**

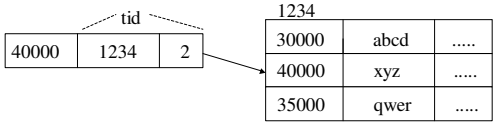
- **tiheä / harva**

- **perushakemisto / ryvästävä h. / oheishakemisto**

Hakemistorakenteet, yleistä - 3

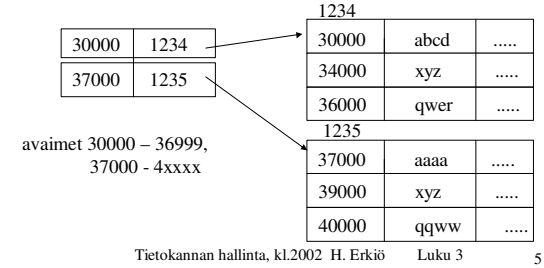
Tiheä (dense) hakemisto: jokaiselle tietueelle on oma hakemistotietue.
Hakemistotietueessa on joko jakso-osoite tai tietueosoite.

- tietueosoite **tid** ('tuple identifier') on muotoa (sivunumero, tietuenumero).



Hakemistorakenteet, yleistä - 3

Harva (non-dense, sparse) hakemisto: hakemistotietue identifioi vain sivun eli p on sivunumero.
Hakemistotietueita on siis vähemmän kuin perustietueita.



Hakemistorakenteet, yleistä - 4

Yleistä hakemistoideasta:

- harvassa hakemistossa hakemistotietueita on vähemmän kuin perustietueita
- hakemistotietue on yleensä perustietuetta lyhyempi (harvassa ja tiheässä hakemistossa)

→ koko tiedosto voidaan hakemiston avulla 'kuvata' perustiedostoa pienemmässä tilassa (keskimäärin vähemmän levyhakuja)

- hakemisto on itsekin levytiedosto (joukko jaksoja)
- hakemistoa tai sen osia pyritään pitämään keskusmuistipuskurissa, jos tilaa riittää (vähentää edelleen levyhakuja)

3.1 Yksitasoiset hakemistot

= perustiedosto + (yksinkertainen) hakemistotiedosto
(hakemistolla ei ole omaa hakemistoaan ...)

Esimerkki. Relaatio emp(ssn, name, salary)

harva hakemisto:

name	sivunro
Aalto	p1
Eerola	p2
Ketola	p3

perustiedosto:

	ssn	name	dno
p1:	5588	Aalto	2
	9595	Aaltonen	3
	. . . . .		
p2:	7777	Eerola	2
	1122	Frank	1
	3333	Hakala	4
p3:	1111	Ketola	1
	4242	Vainio	1

Yksitasoiset hakemistot - 2

perustiedosto:

	ssn	name	dno
p1:	5588	Aalto	2
	9595	Aaltonen	3
. . . . .			
p2:	7777	Eerola	2
	1122	Frank	1
	3333	Hakala	4
p3:	1111	Ketola	1
	4242	Vainio	1

tiheä hakemisto:

ssn	tid
1111	(p3,1)
1122	(p2,2)
3333	(p2,3)
4242	(p3,2)
5588	(p1,1)
7777	(p2,1)
9595	(p1,2)

Yksitasoiset hakemistot - 3

harva hakemisto:

name	sivunro
Aalto	p1
Eerola	p2
Ketola	p3

perustiedosto:

	ssn	name	dno
p1:	5588	Aalto	2
	9595	Aaltonen	3
. . . . .			
p2:	7777	Eerola	2
	1122	Frank	1
	3333	Hakala	4
p3:	1111	Ketola	1
	4242	Vainio	1

tiheä hakemisto:

ssn	tid
1111	(p3,1)
1122	(p2,2)
3333	(p2,3)
4242	(p3,2)
5588	(p1,1)
7777	(p2,1)
9595	(p1,2)

Harva hakemisto on tässä perushakemisto,
tiheä hakemisto oheishakemisto.

Hakemiston käyttö

Tietueen **haku** on kaksivaiheista:
1) etsitään hakemistotietue ja siitä sivun osoite,
2) haetaan perustiedoston sivu.

→ hakemiston käyttö johtaa aina vähintään kahteen levyhakuun, ellei hakemisto ole puskurissa

Hakemistoa käsitellään jaksoittain (ensin etsitään jakso, sitten tietue).

- useita levyhakuja, jos joudutaan todella etsimään
- jos hakemisto on järjestyksessä, esim. binäärihaku käy
- hakemistojaksoja yleensä vähemmän kuin perustiedoston jaksoja
  → käsittely perustiedostoa nopeampaa

Hakemiston käyttö - 2

Tietueen **lisäys** tiedostoon:
- haetaan tietueen paikka (kasassa loppuun, järjestetyssä paikalleen)
- lisätään tietue perustiedostoon
- päivitetään hakemisto(t) uuden tilanteen mukaisiksi

Esim. insert into emp values ('6666', 'Eskola', 2):
- monikko sivulle p2
- tietue ('6666', (p2,4)) tiheään hakemistoon
  (tai (p2,2), jos siirretään tietueita)

Harvaa hakemistoa ei ole tarpeen päivittää eikä näin yleensä tehdä.

Hakemiston käyttö - 3

Tietueen **poisto**: poisto perustiedostosta ja tiheästä hakemistosta ('oikeasti' tai poistomerkintä)

- tietueen päivitys: harj.teht.
  (samoin: operaatioiden levyhakumäärät ...)

Yleisesti: mitä enemmän hakemistoja on, sitä raskaampia ovat perustiedoston muutoksia sisältävät operaatiot.
(vrt. oheishakemistot)

Harva hakemisto

Harvan hakemiston ominaisuudet:

- perustiedoston on oltava hakemistoavaimen mukaisessa järjestyksessä
- hakemistotietue (v, p_i) osoittaa siihen sivuun, jonka ensimmäisen tietueen avainarvo on v; yleisesti on voimassa:
  1) v <= sivun p_i avainarvot
  2) v > sivun p_{i-1} avainarvot (*)
- avainarvo v voi olla muukin ehdot (*) täyttävä avainarvo kuin tiedostossa esiintyvä: ensimmäiseen hakemistotietueeseen sijoitetaan usein arvo "-∞"
- jos perustietueet ketjutetaan, hakemisto voi olla vielä 'harvempi': ei hakemistotietuetta jokaiselle sivulle

Harva hakemisto - 2

Seurauksia:

- useimmat avainarvot eivät esiinny hakemistossa ollenkaan

- tiedostoon voi liittyä vain yksi harva hakemisto kerrallaan (avainten järjestys!)

Tiheä hakemisto

Tiheän hakemiston ominaisuudet:

- perustiedoston järjestys voi olla mikä tahansa
→ tiheä hakemisto tehdään tyypillisesti jonkin muun kuin relaation avainkentän mukaan
  (avainarvon toistuminen vaatii jonkin ratkaisun)
- tiheitä hakemistoja voi olla useita samanaikaisesti
- hakemiston ja perustiedoston sama järjestys vaikuttaa joidenkin hakujen tehokkuuteen (avainarvojen osaväli)
- jokaisella avainarvolla on oma hakemistotietue

Hakemistotyypit

Perushakemisto eli ensisijainen hakemisto (**primary index**)

- tiedoston perusavaimeen (primary key) perustuva hakemisto voi olla harva tai tiheä

- perustiedosto on avainjärjestyksessä
- hakemisto on avainjärjestyksessä

Esim. name-hakemisto edellä (mikäli name on relaation avain).

Hakemistotyypit - 2

Ryvästävä eli järjestävä hakemisto (clustering index)

- perustuu johonkin muuhun kuin relaation avainkenttään, ts. arvojen ei tarvitse olla yksikäsitteisiä
  kenttä = ryvästävä kenttä

- perustiedoston tulee olla järjestyksessä: peräkkäiset samat avainarvot muodostavat rypään (cluster)

- rypääseen kuuluvat tietueet ovat yleensä samassa jaksossa

- hakemisto voi olla harva tai tiheä

(Kirjallisuudessa hieman erilaisia määritelmiä:
onko ryvästävä kenttä yliavain vai ei).)

Hakemistotyypit - 3

Esim. osastoittain perustiedostossa olevat employee-monikot;
ryvästävä kenttä dno:

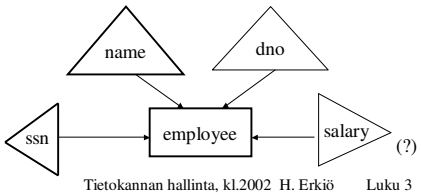
dno	sivunumero	perustiedosto (vain dno näkyy):
1	p1	p1: … , 1; …, 1; … , 1; … , 2
2	p1	p2: … , 2; …, 2; … , 2; … , 2
3	p3	p3: … , 2; …, 2; … , 3, … , 5
5	p3	p4: … , 5 ; . . . .
…	…	

- monia toteutustapoja, esim.
  - jaksoon vain yhden avainarvon tietueita
  - jaksojen linkitys (kuvassa dno = 2 vaatii jotain linkitystä tms.)

Hakemistotyypit - 4

Oheishakemisto = secondary index, toisiohakemisto; muodostaa toissijaisen saantipolun

- ei ole perushakemisto eli ei voi nojautua perustiedoston järjestykseen → hakemisto on välttämättä tiheä
- normaali tapa järjestää useita hakemistoja relaatiolle:



Hakemistotyypit - 5

Oheishakemisto muodostaa loogisen järjestyksen tiedostolle hakemistoavaimensa mukaan.

Tiedoston läpikäynti oheishakemiston mukaisessa järjestyksessä voi olla hidasta; peräkkäiset tietueet voivat olla aina eri sivuilla - hakemistoon kylläkin sopii esim. binäärihaku ...

Jos oheishakemiston hakemistoavain ei ole yksikäsitteinen, on eri vaihtoehtoja:

1. monta hakemistotietuetta / avainarvo

Esim.

dno	tid
1	p2,2
1	p3,1
1	p3,2
2	p1,1
. . .	. . .

2. yksi hakemistotietue, jossa lista osoitteita (vaihtuvanmittainen)

3. hakemistotietue osoittaa erilliseen osoitelistaan, jossa sama-avaimisten tietueiden osoitteet peräkkäin omana jaksonaan:

hakemisto dno	bid	osoitelistat	perustiedosto		(dno)
1	b1	b1: (p2,2)	p1: …	…	2
2	b2	(p3,1)	…	…	3
3	b3	(p3,2)			
4	b4	b2: (p1,1)	p2: …	…	2
…	…	(p2,1)	…	…	1
			…	…	4
		b3: (p1,2)			
			p3: …	…	1
		b4: (p2,3)	…	…	1

3.2 Monitasoiset hakemistot

Hakemisto on tiedosto, jonka rakenne voi olla periaatteessa mikä tahansa. Käsittelyn kannalta sopiva rakenne on usein järjestetty peräkkäistiedosto.

Oikein pienelle hakemistolle riittää vaikka kasarakenne …

Kun hakemisto on suuri, haku peräkkäistiedostosta voi olla liian hidasta.

Järjestetylle hakemistotiedostolle voidaan muodostaa oma (järjestetty) hakemisto; jos tämäkin on liian 'pitkä', sille edelleen oma (järjestetty) hakemisto jne. Näin saadaan monitasoinen hakemistorakenne, jota sanotaan **ISAM-rakenteeksi** (Indexed Sequential Access Method).

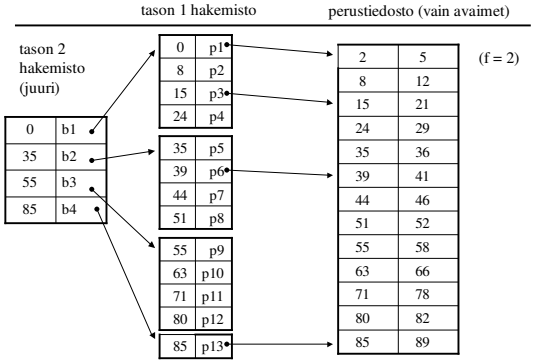
3.2 Monitasoiset hakemistot - 2

ISAM-rakenteen ominaisuuksia:

- Hakemisto on yleensä harva (joten seuraavan tason hakemisto on paljon edellistä pienempi).
- Uusia tasoja rakennetaan, kunnes koko uusi hakemisto mahtuu yhdelle sivulle.

  Käytännössä tasoja ei synny monta, ja ylimmän tason hakemisto(sivu) pysyy yleensä puskurissa.
- Hakemistoja ei päivitetä luonnin jälkeen; hakemistosivuja ei siis tarvitse kirjoittaa takaisin levylle.

Monitasoiset hakemistot – 3 : ISAM-rakenne



3.2 Monitasoiset hakemistot - 4

Esimerkissä toteutuvat harvan hakemiston ominaisuudet (s. 13):
- hakemistoavain = 'viitta' seuraavalle tasolle; peräkkäisiä hakemistoavaimia vertaamalla selviää, mikä hakemistoalkio osoittaa tiettyyn seuraavan tason jaksoon

Sanotaan, että n-alkioisen hakemistosivun P tietue (vi,pi) peittää avainarvon v, jos

(1) i < n ja vi <= v < v(i+1), tai
(2) i = n ja vi <= v.

Huom. Kun hakemistosivuja ei päivitetä, esim. tietuepoistojen jälkeen hakemistoavaimina voi olla aivan muita arvoja kuin perustiedoston avaimet. Rakenne toimii silti ...

3.2 Monitasoiset hakemistot - 5

Avainarvolla v varustetun tietueen haku l-tasoisesta ISAM-rakenteesta:

```
P ← rakenteen juurisivun osoite;
for j ← l downto 1 do
        B:=bufferfix(P);
        etsi puskurisivulta B avainarvon v peittävä
                hakemistotietue (vi, pi);
        P←pi;
        bufferunfix(B);
end;
B:=bufferfix(P);
hae avainarvolla v varustettua tietuetta puskurisivulta B;
tulosta tietue (tms.);
bufferunfix(B);
end.
```


3.2 Monitasoiset hakemistot - 6

Algoritmi vaatii enintään l+1 levyhakua (vähemmän, jos ainakin juuri on jo puskurissa).

Tietueen lisäys: haetaan tietuetta edellisen algoritmin mukaan; lisätään sitten siihen jaksoon, johon avaimensa perusteella kuuluisi. (Jos on avainrajoite ja tietue löytyy, ei tietenkään lisätä …)

Tilan loppuessa perustiedoston jaksosta siihen voidaan ketjuttaa ylivuotojaksoja; tämä ei näy hakemiston puolella, mutta kasvattaa tietysti käsittelyaikaa yhdellä tai useammalla levyhaulla perustiedostossa.

Tasojen lukumäärä?
- määräytyy hakemistotason tiedoston jaksotuskertoimen (eli hakemistojakson koon ja hakemistotietueen pituuden) mukaan

3.2 Monitasoiset hakemistot - 7

Esim. tiedostossa N = 50000 sivua;
avainkentän pituus 12 tavua, sivunumero 4 tavua
ja sivun koko 4 KB.

- hakemistotietue vie enintään 12 + 4 + ? = 20 tavua
- hakemistotietueita mahtuu sivulle 4000/20 = 200 kpl
- tason 1 hakemistosivuja tarvitaan 50000/200 = 250 kpl
- tason 2 hakemistotietueet (250 kpl) mahtuvat kahdelle sivulle
-tasolla 3 kaksi sivuosoitetta mahtuu (oikein hyvin …) yhdelle sivulle (joka on hakemiston juuri)

Yleisesti: tasojen lkm = $\lceil \log_f N \rceil$, missä N = tason 1 tietueiden määrä ja f hakemistosivujen jaksotuskerroin.
Kun yleensä f >>2, hierarkiasta tulee aina hyvin matala.

(3.2) ISAM-rakenteen luonnin vaiheet

1. Järjestetään alkuperäisen tiedoston perustietueet avaimen mukaan nousevaan järjestykseen.

2. Sijoitetaan perustietueet avaimen mukaiseen järjestykseen perustiedostolle varatuille sivuille p1, … , pN. Jätetään kasvunvaraa esim. (jopa) 50 % jokaiselle sivulle.

3. Muodostetaan 1. tason hakemistotietueet (v1, p1, … , (vN, pN) ottamalla vi:n arvoksi pienin sivun pi avainarvo.
Poikkeus: ensimmäiseen tietueeseen kannattaa asettaa vi = pienin mahdollinen arvo ("-∞"), jotta tätäkään hakemistotietuetta ei tarvitse koskaan päivittää.

Sijoitetaan hakemistotietueet sivuille avaimen mukaisessa järjestyksessä (sivut täyteen, koska niitä ei päivitetä).

4. Muodostetaan ylemmät tasot vastaavasti.

ISAM – luonti (jatkoa)

ISAM-rakenteen luonti 'tyhjästä' peräkkäisillä lisäyksillä olisi työlästä:
perustietueet tulee saattaa järjestykseen, ja
rakenne joutuisi hyvin todennäköisesti nopeasti epätasapainoon (pitkiä ylivuotoketjuja).
(Hakemistoa tietysti päivitettäisiin tässä tapauksessa koko ajan tai ainakin ajoittain.)

Normaalitilanteessakin pitkien ylivuotoketjujen syntymisen vaara on olemassa. Rakenteen 'korkeus' onkin oikeastaan l + pisimmän ylivuotoketjun pituus.

Tällöin suoritetaan rakenteen uudelleenorganisointi:
- luetaan vanhasta tiedostosta perustietueet avainjärjestyksessä,
- kirjoitetaan tietueet uudeksi perustiedostoksi (esim. 50 %:n täyttöasteella) ja muodostetaan uusi hakemisto

Hakemistotasojen määrää voidaan kontrolloida luontivaiheessa mahdollisesti valitsemalla jakson koko sopivasti.

Hakemisto on puurakenteena yleensä ‘epätasapainoinen’ (vrt. laskelma sivulla 28): kun hakemistotasoa kasvatetaan luonnissa jaksoittain, tason viimeinen jakso voi olla hyvin vajaa.

Erikoistapaus: Rakenne oli aikanaan tarkoitettu kaksitasoiseksi levymuistin rakenteen mukaisesti: ylin taso = sylinterihakemisto, alempi taso = urahakemisto (IBM ISAM …)

Tiheä ISAM-hakemisto

ISAM-rakenne voidaan toteuttaa myös osoittamaan kasan (ei-järjestetyn perustiedoston) tietueita. Tällöin tulee kuitenkin luoda rakenteen pohjaksi tiheä hakemisto, jossa perustiedoston avaimet ovat järjestyksessä. Pohjatasosta ylöspäin rakenne on normaali ISAM-hakemisto.

Pohjatason hakemistotietue:
(avainarvo, perustiedoston tietueen (tai jakson) osoite)

- pohjatason täyttöaste kuten ISAM-rakenteen perustiedostossa

Huom. Jos tarvitaan vain avainta, mutta ei tietueen muita kenttiä, hakua ei tarvitse ulottaa kasaan asti (esim. avainrajoitteen kontrolli).

Pohjatason tietueet ovat lyhyitä, joten pohjatasolla tarvitaan paljon vähemmän jaksoja kuin kasassa.

ISAM-rakenteen käyttö

Mihin ISAM-rakenne sopii?

- yleisesti kyselyihin, jotka perustuvat hakemistoavaimen tarkkaan arvoon tai sen alkuosaan:

```
select * from R where A = 'abc';
select * from R where A like 'abc%';
```

- hakemistoavain moniosainen, esimerkiksi suku- ja etunimi (lname, fname), ja ehdot sopivassa järjestyksessä:

- tässä tehokkaita kyselyjä (ehtoja):

```
select * from employee
      where fname = 'John' and lname ='Smith';
  … where lname = 'Smith'
  … where lname like 'Sm%'
  … where fname < 'J' and lname = 'Smith'
```


ISAM-rakenteen käyttö - 2

Tehottomia (hakemisto ei auta):

```
... where fname = 'John' and lname like '%son'

… where fname = 'John' and ssn = '123456789'
```