

Janne Lusua

# **Indeksit ja niiden käyttö MySQL- ja SQL Server -tietokantatuotteissa**

Tietojärjestelmätieteen  
kandidaatintutkielma  
3.4.2009

Jyväskylän yliopisto  
Tietojenkäsittelytieteiden laitos  
Jyväskylä

# TIIVISTELMÄ

Lusua, Janne Petteri

Indeksit ja niiden käyttö MySQL- ja SQL Server -tietokantatuotteissa / Janne

Lusua

Jyväskylä: Jyväskylän yliopisto, 2009

35 s.

Kandidaatintutkielma

Tietokantojen koon yhä kasvaessa on syntynyt tarve järjestää tieto siten, että se on mahdollisimman nopeasti saatavilla. Tähän ongelmaan on ratkaisuna indeksit, joita voidaan verrata kirjan sisällysluetteloon. Tässä tutkielmassa tarkastellaan, minkä tyyppisiä indeksejä on olemassa ja miten niitä hyödynnetään kahdessa suositussa tietokannan hallintajärjestelmässä, MySQL:ssä ja Microsoft SQL Serverissä.

Aihetta käydään tutkielmassa läpi esittelemällä ensin eri indeksityypit. Tämän jälkeen kuvataan tarkempaan tarkasteluun valitut tietokantatuotteet perustasolla ja keskitytään niihin osa-alueisiin, jotka liittyvät indekseihin. Lopuksi vertaillaan samantyyppisten indeksien toteutusta valituissa tietokantatuotteissa.

Tämän kirjallisuuskatsauksen perusteella voidaan todeta, että erilaisia indeksityyppejä on paljon ja niiden määritelmät vaihtelevat hieman lähteestä riippuen. Vertailun perusteella voidaan myös todeta, että samantyyppisten indeksien toteutuksessa on havaittavissa teknisiä eroavaisuuksia eri tietokantatuotteiden välillä.

AVAINSANAT: tietokannan hallintajärjestelmä, indeksit, tietokannan tehokkuus

Ohjaaja: Mauri Leppänen  
Tietojenkäsittelytieteiden laitos  
Jyväskylän Yliopisto

Tarkastaja: Mauri Leppänen  
Tietojenkäsittelytieteiden laitos  
Jyväskylän Yliopisto

# SISÄLLYSLUETTELO

1 JOHDANTO.....	5
2 INDEKSIT.....	7
2.1 Määritelmä.....	7
2.2 Perusindeksi .....	8
2.3 Ei-ryvästetty indeksi.....	9
2.4 Ryvästävä indeksi.....	9
2.5 Toisioindeksi .....	10
2.6 B-puu .....	11
2.7 B <sup>+</sup> -puu.....	12
2.8 R-puu .....	13
2.9 Bittikarttaindeksi .....	15
2.10 Monitasoinen indeksi.....	16
2.11 Teksti-indeksi .....	17
2.12 Hajautusindeksi .....	18
3 INDEKSIT TIETOKANTATUOTTEISSA .....	20
3.1 MySQL.....	20
3.1.1 Tallennusmoottorit.....	20
3.1.2 Indeksien toteutus.....	22
3.2 Microsoft SQL Server .....	24
3.2.1 Tietokantamoottori .....	25
3.2.2 Indeksien toteutus.....	25
3.3 Vertailu.....	29
4 YHTEENVETO .....	31
LÄHDELUETTELO .....	33

## 1 JOHDANTO

Kun tietokantaan lisätään tietoja, tietokannan hallintajärjestelmä tallentaa ne oletuksena siinä järjestyksessä, missä ne syötetäänkin. Kun tietokannasta etsitään tietoa, täytyy jokainen tietokantataulun rivi käydä läpi järjestelmällisesti, että löydetään haluttu tieto. Tällainen menettely voi olla toisinaan hidasta. Tietokannan indeksi poistaa tarpeen järjestelmälliseen läpikäymiseen tarjoamalla kirjan sisällysluettelo vastaaavan, järjestetyn tietorakenteen, josta löytyy hakusanan lisäksi viittaus tietokantataulun tietueeseen, jossa varsinainen tieto sijaitsee. (Connolly & Begg 2005)

Tämän kirjallisuuskatsaukseen perustuvan tutkielman tarkoituksena on tarkastella, minkä tyyppisiä indeksejä on olemassa ja miten niitä hyödynnetään suosituissa tietokannan hallinta-järjestelmissä kuten MySQL:ssä ja Microsoft SQL Serverissä. Tutkimusongelmana on: *Miten eri tietokantatuotteiden indeksit poikkeavat toisistaan?* Tutkimusongelma on jaettu kolmeen tutkimuskysymykseen: *Millaisia indeksejä on kehitetty ja mihin käyttötarkoituksiin? Millaisia indeksejä on toteutettu MySQL- ja Microsoft SQL Server - tietokantatuotteissa? Millaisia eroja samantapaisten indeksien toteutuksissa on havaittavissa ko. tuotteissa?* Tutkimustavoitteena on tuottaa MySQL:n ja Microsoft SQL Serverin indeksien vertailu.

Tutkielma rajataan koskemaan kahta eri tietokantatuotetta, MySQL:ää ja Microsoft SQL Serveriä. MySQL valittiin siksi, että se on ilmainen ja laajasti käytössä oleva tietokantatuote (Sun Microsystems 2009f). Tarkastelun kohteeksi valittiin MySQL:n versio 5.0, joka julkaistiin lokakuussa 2005. Vastineeksi valittiin Microsoft SQL Server, joka edustaa järeää ja maksullista tietokantatuotetta, tarjoten näin hyvän vastakohtan vertailulle. Microsoft SQL Serverin kohdalla tarkasteluun valittiin versio 2005, joka on julkaistu lokakuussa 2005 (Microsoft Corporation 2009).

Tutkielma jakaantuu neljään lukuun. Luvussa 2 kuvataan perusteita indeksien käytölle sekä esitellään erilaiset indeksityypit. Luvussa 3 kuvataan tarkempaan tarkasteluun valitut tietokantatuotteet sekä indeksien toteutukset ko. tuotteissa. Lisäksi esitetään vertailu ko. tietokantatuotteiden indeksien toteutusten välillä. Tuloksena on johtopäätöksiä siitä, missä määrin samantyyppisten indeksien toteutuksessa on eroja tuotteiden välillä. Luku 4 on yhteenveto, jossa kerrataan tutkielman keskeinen sisältö ja tulokset.

## 2 INDEKSIT

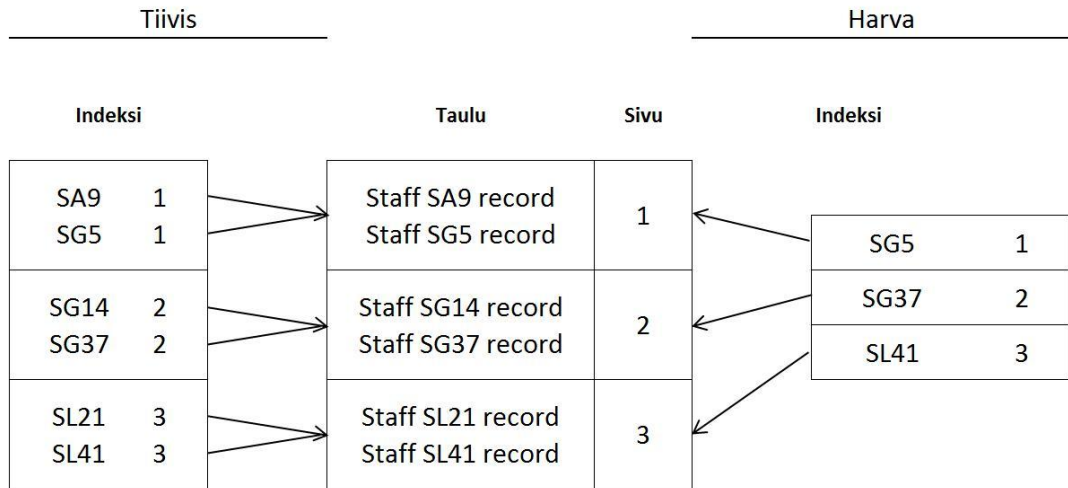
Indeksi on tietokannan tauluun liitetty aputietorakenne, johon voidaan viitata kuten kirjan lukija viittaa sisällysluetteloon etsiessään, millä sivuilla haettu termi ilmenee (Connolly & Begg 2005). Indeksiä voidaankin pitää ratkaisuna ongelmaan: "Miten voidaan pieni määrä tietoa löytää suuresta tietovarastosta ilman, että yksitellen käydään läpi kaikki tietovaraston tietueet?". Voidaan kuvitella esimerkiksi tilanne, että nostettaessa rahaa pankkiautomaatilta järjestelmä suorittaa lineaarisen haun mahdollisesti jopa miljoonien asiakkaiden tietueiden joukosta, kunnes löytää oikean tilin tiedot. Tällainen haku kestäisi luultavasti liian kauan, eikä käytettävyyks olisi enää vaaditulla tasolla. Tällaiset suuret, sähköiset tietokannat ovat kasvattaneet tarvetta erilaisille indeksointimenetelmille, jotta tarvittava tieto saadaan haettua nopeasti ja tehokkaasti jopa yhdellä levyoperaatiolla. Seuraavaksi esitetään indeksin määritelmä sekä eri indeksityypit. (Lightstone, Teorey & Nadeau 2007)

### 2.1 Määritelmä

Indeksi on tietorakenne, jonka avulla tietokannan hallintajärjestelmän on mahdollista hakea tiettyjä tietoja taulusta nopeammin, ja siten nopeuttaa vasteaikaa käyttäjän kyselyille (Connolly & Begg 2005). Indeksi määritellään yleensä niille tietokannan kentille, jotka toistuvat usein haku- tai lajitteluehdoissa. Tämä mahdollistaa sekä yksittäisten hakujen nopeuttamisen että suurtenkin tietomäärien järjestämisen. Määrittely voidaan tehdä käyttäen yhtä tai useampaa saraketta, ja relaatiotietokannoissa indeksi onkin usein kopio osasta tietokantataulua.

Indeksi voi olla toteutukseltaan tiivis (dense) tai harva (nondense, sparse). Tiiviissä toteutustavassa jokaista tietokannan tietuetta kohden löytyy tietue myös indeksitietorakenteesta (Elmasri & Navathe 2004). Harvassa

toteutustavassa on vastaavasti vain osa hakuavainten arvoista (Connolly & Begg 2005). Eri toteutustapojen eroja on havainnollistettu kuviossa 1.



Kuvio 1 Esimerkki tiiviistä ja harvasta indeksistä. Mukailtu lähteestä (Connolly & Begg 2005, 1277).

## 2.2 Perusindeksi

Perusindeksiä (primary index) kutsutaan myös nimellä päähakemisto. Koska perusindeksi perustuu järjestetyn tiedoston fyysiseen tallennusrakenteeseen, voi samassa tietokantataulussa olla vain yksi perusindeksi tai ryvästävä indeksi. (Elmasri & Navathe 2004)

Perusindeksi on järjestetty taulu, joka sisältää kaksi kiinteämittaista kenttää. Ensimmäiseen kenttään on tallennettu taulun järjestysavaimen arvo, joka toimii samalla taulun perusavaimena (primary key). Toinen kenttä sisältää osoittimen levylohkoon, jossa perusavaimen arvoa vastaava tietue sijaitsee. Jokaista levylohkoa kohti on tarkalleen yksi tietue perusindeksitaulussa, mutta kaikkia taulun tietueita kohti ei välttämättä ole olemassa tietuetta perusindeksitaulussa. Pääindeksi on siten toteutukseltaan harva. (Elmasri & Navathe 2004)

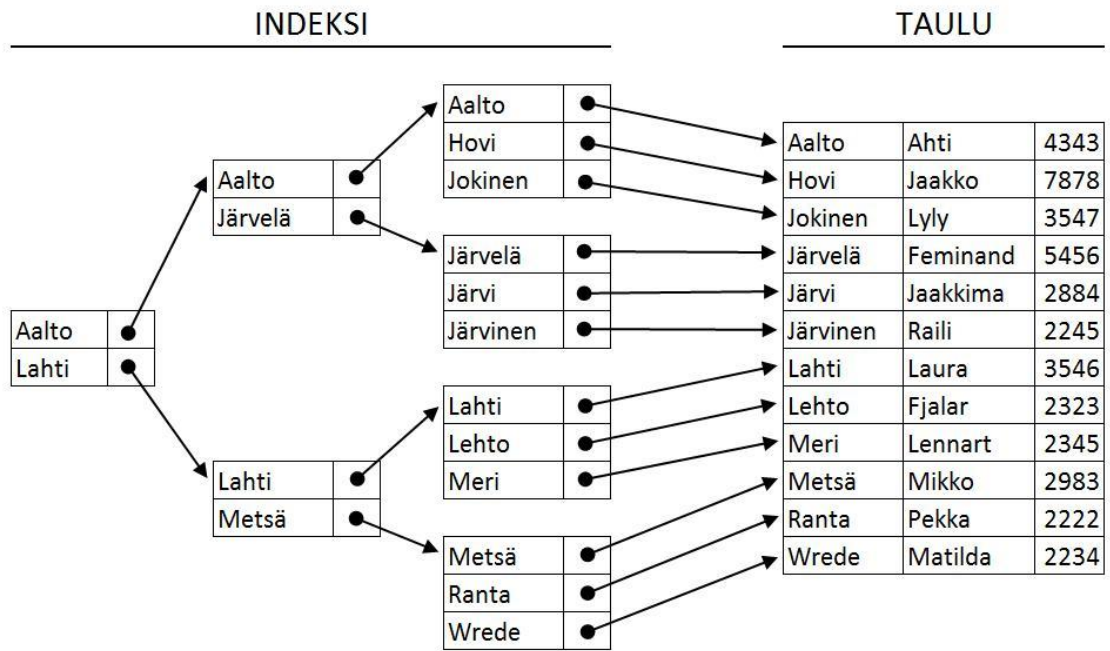


### **2.3 Ei-ryvästetty indeksi**

Ei-ryvästetty indeksi (nonclustered index) on erillinen, rivien fyysisestä järjestyksestä riippumaton indeksirakenne. Tästä syystä niitä voidaan tehdä yhteen tietokantatauluun jopa 249 kappaletta. Kirjan sisällysluettelo -esimerkkiin viitaten ei-ryvästetyn indeksin tapauksessa lukija voi joutua selailemaan useita eri puolilla kirjaa olevia sivuja löytääkseen kaikki viitteet, jotka liittyvät haettuun aiheeseen. Ei-ryvästetty indeksi tallennetaan usein B-puu-tietorakenteeseen, jota kuvio 4 havainnollistaa. (Rankins 2007)

### **2.4 Ryvästävä indeksi**

Ryvästävä indeksi (clustering index) on indeksityyppi, joka vaikuttaa myös taulun rivien järjestykseen. Taulun rivit ovat siis samassa järjestyksessä kuin indeksikin. On tietokannan hallintajärjestelmän tehtävä huolehtia siitä, että tämä järjestys myös säilyy. Tätä järjestystä ja ryvästävän indeksin rakennetta on havainnollistettu kuviossa 2. Koska taulu voidaan järjestää vain yhdellä tavalla kerrallaan, voi sillä olla luonnollisesti vain yksi ryvästävä indeksi. Siksi onkin ensiarvoisen tärkeää valita ryvästävä indeksi oikein. On kuitenkin otettava huomioon, että kaikki markkinoilla olevat tietokantajärjestelmät eivät tue ryvästäviä indeksejä. (Hovi, Huotari & Lahdenmäki 2005)



Kuvio 2 Esimerkki ryvästävästä indeksistä: taulun rivit ovat samassa järjestyksessä kuin indeksi. (Hovi, Huotari & Lahdenmäki 2005, 157)

## 2.5 Toisioindeksi

Toisioindeksi (secondary index) on perusindeksin tapaan järjestetty taulu, jossa on kaksi kenttää (Connolly & Begg 2005). Ensimmäisessä kentässä on taulun jonkin sellaisen kentän arvo, jonka mukaan datatiedostoa ei ole fyysisesti järjestetty. Toisessa kentässä on joko lohko- tai tietueosoitin levyille. Perusindeksistä poiketen yhtä taulua kohti voi olla useita toisioindeksejä. Toisioindeksin toteutustapa on tiivis, joten jokaista taulun tietuetta kohti on oltava tietue myös toisioindeksitiedostossa. Toisioindeksi on mahdollista muodostaa myös sellaiselle taulun kentälle, joka ei kelpaa esimerkiksi avaimeksi toistuvien arvojen vuoksi. Toisioindeksi tarvitsee usein enemmän tallennustilaa ja hakuajat ovat pidemmän kuin perusindeksissä, koska tietueita on enemmän. (Elmasri & Navathe 2004) Toisioindeksiä on havainnollistettu kuviossa 3.

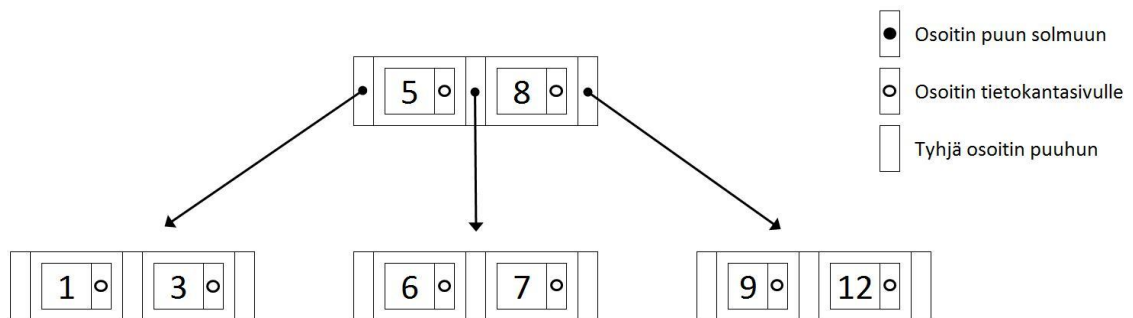
TOISIOINDEKSI		TAULU		
Järjestävän kentän arvo	Osoite	Osoite	Sukunimi	Etunimi
Abdul	0x2000	0x1F00	Sanderson	James
Alex	0x2400	0x2000	Smith	Abdul
Ikir	0x2500	0x2100	Smith	Jill
James	0x1F00	0x2200	Smith	Rajir
Jill	0x2100	0x2300	Sopley	John
John	0x2300	0x2400	Stanhope	Alex
Ptolemy	0x2600	0x2500	Szechl	Ikir
Rajir	0x2200	0x2600	Szechl	Ptolemy

Kuvio 3 Esimerkki toisioindeksistä. Mukailtu lähteestä (Lightenna Limited 2007).

## 2.6 B-puu

B-puu on puutietorakenteeseen perustuva tietorakenne. Puu (tree) koostuu solmuista (node), joista jokaisella paitsi juurisolmulla (root), on yksi emosolmu (parent) ja nolla tai useampi lapsisolmu (child). Juurisolmulla ei ole emosolmuja. Solmuja, jolla ei ole yhtään lapsisolmuja, kutsutaan lehtisolmuiksi (leaf) ja vastaavasti solmuja jotka eivät ole juuri- eivätkä lapsisolmuja, kutsutaan sisäisiksi (internal) solmuiksi. Juurisolmun taso on nolla, ja muiden solmujen taso on aina yhden suurempi kuin niiden emosolmun taso. Arvojen lisäämiseen ja poistamiseen puusta tarvittavat algoritmit eivät yleisesti takaa, että puu on tasapainotettu (balanced). Tasapainotettu tarkoittaa tässä tapauksessa sitä, että kaikki puun lehtisolmut ovat samalla tasolla. Connollyn ja Beggin (2005) mukaan tasapainotettu tarkoittaa, että jokainen polku juurisolmusta lehtisolmuun on yhtä pitkä. B-puuhun liittyy kuitenkin ylimääräisiä rajoitteita (constraint), jotka takaavat sen, että b-puu on aina tasapainotettu ja esimerkiksi tietoja poistettaessa "hukattu" tila ei kasva kohtuuttomaksi. (Elmasri & Navathe 2004)

B-puun solmujen täyttöaste pidetään 50 ja 100 prosentin välillä, josta seuraa että jokainen solmu on ainakin puoliksi täynnä. Mikäli esimerkiksi arvojen poiston yhteydessä solmun täyttöaste menee alle 50 prosenttiin, solmu yhdistetään naapurisolmun kanssa. Kaikissa b-puun solmuissa on osoittimet tietokantasivuille, joissa varsinainen tieto sijaitsee. Tämä ilmenee myös kuvioista 4. B-puun muodostuminen alkaa yksittäisestä juurisolmusta tasolla nolla, joka on sillä hetkellä myös lehtisolmu. Kun juurisolmu on täyttynyt ja puuhun lisätään seuraava arvo, juurisolmu jakautuu (split) kahdeksi solmuksi tasolle yksi. Vain keskimääräinen arvo pidetään juurisolmussa, ja loput arvot jaetaan tasaisesti kahdelle muulle solmulle. Kun juurisolmusta jakautunut solmu täyttyy ja siihen lisätään uusi arvo, se jakautuu kahdeksi solmuksi samalle tasolla ja keskimääräinen arvo siirretään emosolmuun, johon lisätään myös osoittimet kahteen syntyneeseen solmuun. Mikäli emosolmu täyttyy, sekin jaetaan. Jakautuminen voi ulottua aina juurisolmuun asti, jonka jakautuessa syntyy uusi taso. Arvojen poistaminen voi myös vähentää tasojen määrää. B-puuta on havainnollistettu kuviossa 4. (Elmasri & Navathe 2004)



Kuvio 4 Esimerkki B-puusta. Arvot on syötetty järjestyksessä 8, 5, 1, 7, 3, 12, 9, 6. (Elmasri & Navathe 2004, 473)

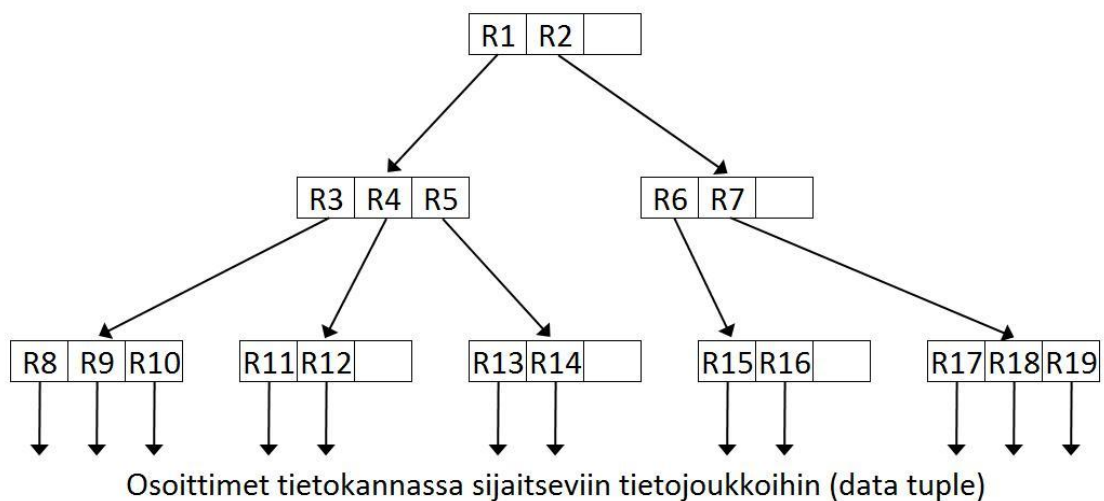
## 2.7 B<sup>+</sup>-puu

B<sup>+</sup>-puu on muunnelmä B-puu-tietorakenteesta, jossa lehtisolmujen rakenne poikkeaa sisäisten solmujen rakenteesta. B<sup>+</sup>-puussa on osoittimia

tietokantasivuille vain lehtisolmuissa. Mikäli hakukenttänä käytetään avainkenttää, on lehtisivuilla tietue ja osoitin tietokantasivulle jokaista hakukentän arvoa kohden. Mikäli hakukenttä ei ole avainkenttä, osoittaa osoitin lohkokon, joka sisältää osoittimet varsinaisille tietokantasivuille. Lehtisolmut linkitetään yleensä toisiinsa, jolla saadaan hakukentälle järjestetty pääsy tietueisiin. (Elmasri & Navathe 2004)

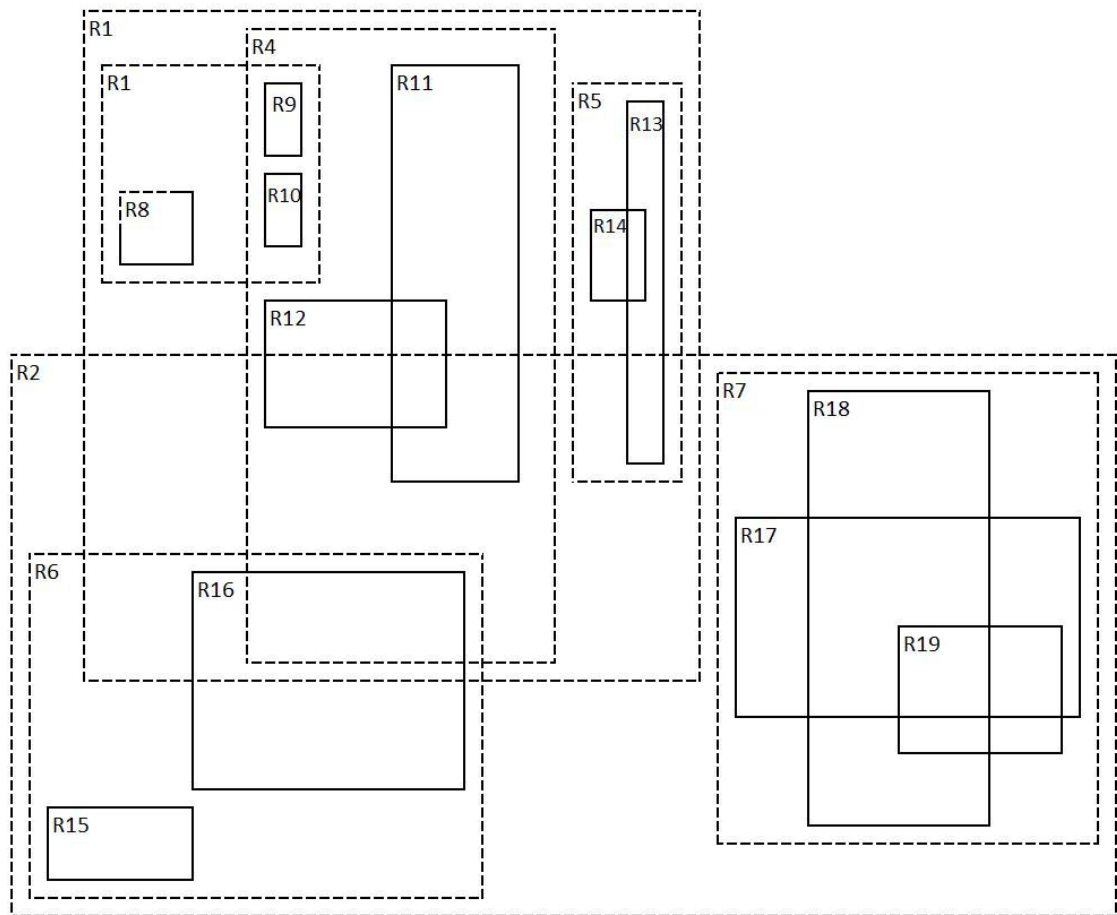
## 2.8 R-puu

R-puu on korkeuden mukaan tasapainotettu (height-balanced) puu-rakenne, jossa lehtisivujen indeksitietueet sisältävät osoittimet tietokantasivuille. Tietueet esitetään väleinä (interval) eri ulottuvuuksissa (dimension). Rakenne on suunniteltu siten, että avaruudellisten (moniulotteiset, spatial) hakujen tarvitsee käydä mahdollisimman vähässä määrässä solmuja. Rakennetta on havainnollistettu kuviossa 5. Kuvioista voidaan havaita esimerkiksi, että suorakulmio R1 sisältää suorakulmiot R3, R4 ja R5, jotka taas sisältävät esimerkiksi suorakulmion R8. Indeksiin kohdistuvat lisäykset ja poistot voidaan suorittaa hakujen yhteydessä, joten erillistä, säännöllistä uudelleenorganisointia ei tarvita. Indeksi on siten täysin dynaaminen. (Guttman 1984)



Kuvio 5 R-puun rakenne (Guttman 1984)

Yksi yleinen avaruudelliselle tiedolle suoritettava operaatio on etsiä kaikki tietyllä alueella sijaitsevat kohteet, esimerkiksi: *Hae kaikki kunnat, jotka omistavat maata 20 kilometrin säteellä tietyistä pisteistä*. Tämän tyyppisiä hakuja tarvitaan usein esimerkiksi tietokoneavusteisessa suunnittelussa (CAD). Klassiset yksiulotteiset tietokantojen indeksirakenteet eivät sovellu tällaisiin moniulotteisiin avaruudellisiin hakuihin, koska hakuavaruus on moniulotteinen. R-puussa suorakulmioiden välillä voi olla myös päällekkäisiä suhteita, jota havainnollistetaan kuviossa 6. Esimerkiksi suorakulmiolla R12 on päällekkäisiä alueita suorakulmioiden R1, R2 ja R11 kanssa. (Guttman 1984)



**Kuvio 6 R-puun suorakulmioiden päällekkäisyys (Guttman 1984)**

## 2.9 Bittikarttaindeksi


Lightstonen, Teoreyn & Nadeaun (2007) mukaan bittikarttaindeksi (bitmap index) on B-puu-indeksiä uudempi indeksityyppi, joka on tarkoitettu monen hakuehdon kyselyihin. Tehokkaimmillaan bittikarttaindeksi on haettaessa vaikkapa ehdot täyttävien rivien määrää tai kun taulusta haettavia rivejä on suhteellisen vähän. Myös tilantarve on vähäinen, sillä bittikarttaindeksit ovat yleensä tiivistettyjä. Bittikarttaindeksit nopeuttavat erikoisia indeksointioperaatioita kuten leikkaus (intersection) ja yhdiste (union). Varjopuolena sopivuus operatiivisiin toimintoihin on huono, sillä indeksiä rakennettaessa koko taulu on lukittava ja näin estettävä sen käyttö. Tämä ei ole käytännössä mahdollista esimerkiksi pankkitilien tapahtumia hoitavassa järjestelmässä.

Rakenteeltaan bittikarttaindeksi poikkeaa esimerkiksi B-puu-indeksistä. Jokaista tietokantataulun riviä edustaa yksittäinen bitti bittivektorissa, ja jokaisella ehdon täyttävällä rivillä on oma bittinsä asetettu tähän samaan bittivektoriin. Täten jokaisella attribuuttiarvolla on oma erillinen bittivektorinsa. Ehdot täyttävät rivit löydetään suorittamalla AND -operaatio bittivektoreille. (Lightstone, Teorey & Nadeau 2007)

Hovi, Huotari ja Lahdenmäki (2005) toteavat: *Indeksoitavan sarakkeen kutakin erilaista arvoa kohti syntyy oma bittivektori. Jos esimerkiksi sukupuolikoodi on indeksoitu, syntyy kaksi bittivektoria, toinen miehille, toinen naisille. Mies-bittivektorissa on ykkönen niillä kohdilla, joissa taulussa on mies. Eli jos taulun kolmas rivi on mies, on bittivektorin kolmas arvo ykkönen, muutoin siis nolla.* Kuviossa 7 on havainnollistettu bittikarttaindeksointiin perustuvaa hakua.

Kysely: Montako miestä Rovaniemellä on vakuutettu?

Sukupuoli	Vakuutettu	Kunta
M	E	Hki
M	K	Roi
N	E	Tre
M	K	Roi



Sukupuoli = M	Vakuutettu = K	Kunta = Roi
1	0	0
1	1	1
0	0	0
1	1	1

Tulos = 2

**Kuvio 7** Bittikarttaindeksointiin perustuva haku (Hovi, Huotari & Lahdenmäki 2005, 155)

Edellä olevassa kuviossa sarakkeet Sukupuoli ja Vakuutettu vievät vähän tilaa, sillä arvot ovat 1 tai 0. Jokaista erilaista kuntaa kohti tulee oma bittivektori. Tällainen haku on hyvin nopea, sillä bittivektorit mahtuvat hyvin keskusmuistiin ja bitin "AND"-operaation suorittaminen on nopeaa. (Hovi, Huotari & Lahdenmäki 2005)

Hovin, Huotarin ja Lahdenmäen (2005) mukaan bittikarttaindeksi on parhaimmillaan sarakkeelle, jonka arvojoukko on suhteellisen pieni. Esimerkiksi kunta-sarake käy vielä hyvin, sillä kuntia on 348 (23.3.2009) eli syntyy 348 bittivektoria. Henkilötunnus-sarake vastaavasti soveltuu huonosti bittikarttaindeksointiin, koska samanlaisia henkilötunnuksia ei pitäisi olla.

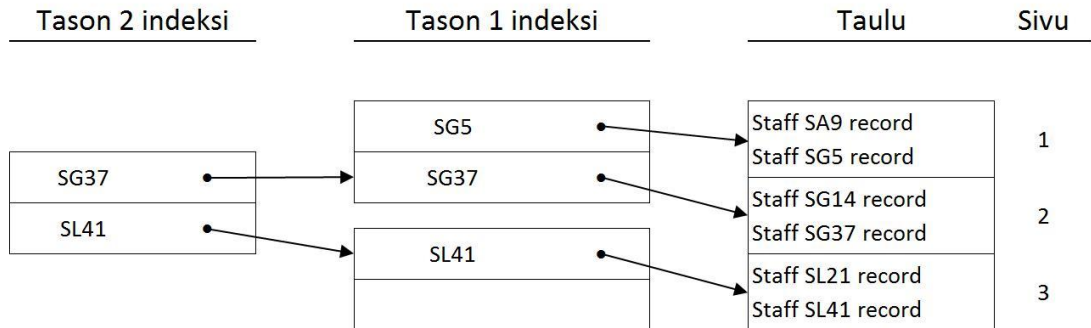
Nykypäivänä bittikarttaindeksi on tuettuna ainakin johtavissa tietokannan hallintajärjestelmissä kuten DB2, Oracle ja Microsoft SQL Server. (Lightstone, Teorey & Nadeau 2007)

## 2.10 Monitasoinen indeksi

Monitasoisen indeksin (multilevel index) tarkoitus on supistaa hakujoukkoa jakamalla indeksi useisiin pienempiin indekseihin ja näin ylläpitää indeksiä



indeksistä. (Connolly & Begg 2005) Monitasoista indeksiä on havainnollistettu kuviossa 8.



**Kuvio 8 Esimerkki monitasoisesta indeksistä (Connolly & Begg 2005, 1280).**

Jos esimerkiksi halutaan etsiä tietue, jossa ilmenee merkkijono SG14, aloitetaan haku tason 2 indeksistä ja haetaan arvon SG14 lähintä arvoa, eli tässä tapauksessa SG37. Tämä tietue sisältää osoitteen tason 1 indeksiin, josta hakua voidaan jatkaa. Toistettaessa edellä mainittu prosessi päädytään taulun sivulle 2, josta haettu tietue löytyy. (Connolly & Begg 2005)

## 2.11 Teksti-indeksi

Teksti-indeksi on indeksityyppi, joka on tarkoitettu sarakkeessa olevan tekstin hakuun kokonaisuudessaan tai vain osaan tekstistä. Teksti-indeksin avulla voidaan esimerkiksi hakea kahta toisiaan lähellä olevaa sanaa tekstistä, perittyä tai johdettua sanaa (esimerkiksi run, ran, running) tai sanaa joka vastaa tai lähes vastaa haettua sanaa. (Gunderloy 2004)

Kuviossa 9 on havainnollistettu teksti-indeksin rakennetta. Ensin on kuvattu osa tietokannan Dokumentti-aulusta. Sarake "DokumenttiID" kuvaa dokumentin yksikäsitteistä tunnistetta ja sarake "Otsikko" vastaavasti dokumentin otsikkoa. Tämän jälkeen on kuvattu teksti-indeksin rakenne, joka on luotu Dokumentti-aulun Otsikko-sarakkeelle. Sarake "Avainsana" sisältää yksittäisiä sanoja (token), jotka on purettu Otsikko-kentän sisällöstä tiettyjä

sääntöjä käyttäen. Sarake "SarakeID" sisältää arvon, joka vastaa tiettyä taulua ja saraketta, joille kyseessä oleva teksti-indeksi on luotu. "DokumenttiID" -sarake sisältää kokonaisluvun, joka viittaa tiettyyn tekstiavaimen arvoon taulussa, jolle indeksi on luotu. "Esiintyminen" -sarakeessa on kokonaisluku, joka kuvaa tietyn hakusanan vastaavuutta kunkin dokumentin kohdalla. Tätä arvoa voidaan käyttää esimerkiksi dokumenttien pisteiden laskemisessa. (Microsoft Corporation 2009)

Osa tietokannan Dokumentti-taulusta

DokumenttiID	Otsikko
1	Crank Arm and Tire Maintenance
2	Front Reflector Bracket and Reflector Assembly 3
2	Front Reflector Bracket Installation

Osa Otsikko-sarakkeelle luodusta teksti-indeksistä

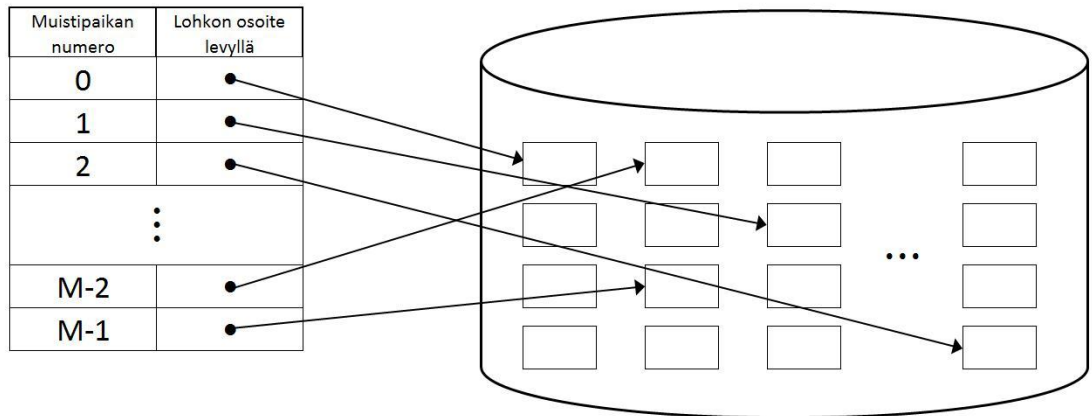
Avainsana	SarakeID	DokumenttiID	Esiintyminen
Crank	1	1	1
Arm	1	1	2
Tire	1	1	4
Maintenance	1	1	5
Front	1	2	1
Front	1	3	1
Reflector	1	2	2
Reflector	1	2	5
Reflector	1	3	2
Bracket	1	2	3
Bracket	1	3	3
Assembly	1	2	6
3	1	2	7
Installation	1	3	4

**Kuvio 9** Esimerkki teksti-indeksin rakenteesta (Microsoft Corporation 2009, [http://technet.microsoft.com/en-us/library/ms142505\(SQL.90\).aspx](http://technet.microsoft.com/en-us/library/ms142505(SQL.90).aspx))

## 2.12 Hajautusindeksi

Hajautusmenetelmän perimmäinen idea on tarjota satunnaisfunktio  $h$ , joka suoritetaan tietueen kentän hajautusarvolle. Funktio tuottaa muistipaikan (bucket) osoitteen, missä tietue sijaitsee. Lohkon sisällä tapahtuva tietueen haku voidaan siten suorittaa keskusmuistin puskurissa. Menetelmää käyttäen tarvitaan vain yksi haku (access) lohkoon useimpien tietueiden hakemiseksi. Hyvän hajautusfunktion tavoitteena on jakaa tietueet tasaisesti koko käytössä olevalle alueelle ja minimoida törmäykset. (Elmasri & Navathe 2004)

Levyllä sijaitsevien tiedostojen hajautusta kutsutaan ulkoiseksi hajautukseksi (external hashing). Jotta menetelmä sopii levyille tallennukseen, kohdeosoittimet osoitetaan muistipaikkoihin (bucket), jotka voivat sisältää useita tietueita. Muistipaikalla tarkoitetaan tässä tapauksessa joko yhtä levylohkoa tai vierekkäisten lohkojen ryvästä (cluster of contiguous blocks). Hajautusfunktio luo viittauksen avaimesta sitä vastaavaan muistipaikan numeroon, eikä siis muistipaikan lohko-osoitteeseen. Muistipaikkojen numerot muutetaan vastaavien levylohkojen osoitteiksi taulukossa, jota ylläpidetään tiedoston otsikkotiedoissa (header). Tätä rakennetta on havainnollistettu kuviossa 10. (Elmasri & Navathe 2004)



Kuvio 10 Muistipaikan numeron täsmäys levylohkon osoitteeseen. (Elmasri & Navathe 2004, 437)

### 3 INDEKSIT TIETOKANTATUOTTEISSA

Tässä luvussa esitellään vertailuun valitut tietokantatuotteet pintapuolisesti sekä esitetään tarkempi kuvaus näiden tukemista indeksityypeistä ja niihin vaikuttavista asioista kuten tallennusmoottorit. Lopuksi esitetään vertailu samantapaisten indeksityyppien välillä näissä tietokantatuotteissa.

#### 3.1 MySQL

MySQL on SQL-standardiin (Structured Query Language) pohjautuva tietokantojen hallintajärjestelmä, jonka ensimmäinen versio julkaistiin vuonna 1995 ruotsalaisen MySQL AB:n toimesta. Sun Microsystems osti MySQL AB:n 16. tammikuuta 2008 (Sun Microsystems 2009d). MySQL-järjestelmä on kaksoislisensoitu. Käyttäjät voivat halutessaan käyttää järjestelmää avoimen lähdekoodin tuotteena GNU GPL -lisenssin alaisuudessa tai ostaa kaupallisen lisenssin MySQL AB:lta. (Sun Microsystems 2009a)

MySQL-tietokannan hallintajärjestelmä on käytössä mm. NASA:lla (National Aeronautics and Space Administration) julkisten sopimusten tallennuksessa (Sun Microsystems 2009c). Myös useat suositut Internet-sivustot kuten LinkedIn, Flickr ja Wikipedia käyttävät palveluissaan MySQL-järjestelmää (Sun Microsystems 2009b).

##### 3.1.1 Tallennusmoottorit

MySQL tarjoaa sovellusten kehittäjille mahdollisuuden valita, mitä tietojen tallennusmoottoria (storage engine) halutaan käyttää. Seuraavaksi esitellään neljä yleisintä ja eniten käytettyä tallennusmoottoria lähteiden (MySQL AB 2004) ja (Sun Microsystems 2009e) mukaan.

Oletustallennustapana käytetään tiedostopohjaista MyISAM-tallennusmoottoria, jossa tietokannat tallennetaan hakemistoihin, ja tietokantataulut näiden hakemistojen alle kolmeen eri tiedostotyyppiin. Erillisiin tiedostotyyppisiin tallennetaan tietokantataulun rakenne eli kaava, varsinainen tieto ja indeksit. MyISAM-tallennusmoottoriin sisältyy kolme eri indeksointimenetelmää: BTREE, RTREE ja FULLTEXT. Yleisimmin käytetty on BTREE-menetelmä. RTREE-menetelmää käytetään indeksoimaan maantieteellistä (GIS) tietoa ja FULLTEXT-menetelmä on suunniteltu erityisesti MySQL full-text search -järjestelmän käyttöön. (MySQL AB 2004)

InnoDB-tallennusmoottori tukee peruutusta (rollback) ja vikatilanteista toipumista (crash-recovery) tietojen suojelemiseksi. InnoDB tallentaa tiedon ryvästävään indeksiin (clustered index). Kaikki InnoDB:n indeksit ovat B-puu-tietorakenteeseen perustuvia, joissa osoittimet on tallennettu puun lehtisivuille. Indeksisivu on oletuksena kooltaan 16 KB. Kun indeksiin lisätään uusia tietueita, tallennusmoottori pyrkii jättämään 1/16 lehden tilasta vapaaksi tulevia lisäyksiä ja päivityksiä varten. InnoDB on suunniteltu mahdollisimman tehokkaaksi käytettäessä suuria tietomääriä. Tieto ja indeksit tallennetaan "taulualueeseen" (tablespace), joka voi koostua useista tiedostoista tai levyosioista. InnoDB käyttää BTREE-indeksejä järjestetyllä perusavaimella. Tallennusmoottori osaa luoda automaattisesti myös hajautusindeksejä (hash index) keskusmuistiin, mikäli se tunnistaa yleisiä kyselyrakenteita. (MySQL AB 2004, Sun Microsystems 2009e)

MEMORY-tallennusmoottori tunnettiin aikaisemmin nimellä HEAP. MEMORY tallentaa kaiken tiedon keskusmuistiin, ja on täten todella nopea. Keskusmuistia vaaditaan kuitenkin riittävästi, ja kaikki tieto häviää esimerkiksi palvelimen uudelleenkäynnistyksen yhteydessä. Taulun rakenne kuitenkin säilyy, koska se tallennetaan levyille. MEMORY-tallennusmoottori tukee HASH- ja BTREE-indeksejä. Yhdessä taulussa voi olla korkeintaan 32 indeksiä, yhdessä

indeksissä voi olla korkeintaan 16 saraketta ja avaimen maksimipituus on 500 tavua. (Sun Microsystems 2009e)

NDB (käytetään myös nimeä NDBCLUSTER, NDB tulee sanoista Network Database) on MySQL Clusterin käyttämä tallennusmoottori, joka mahdollistaa klusteroinnin. Klusteri on Wikipedian (2009a) mukaan useamman tietokoneen verkotettu malli, jossa yleensä yksi tietokone, joka toimii palvelimena, jakaa muiden tietokoneiden eli solmujen (node) kesken tehtäviä. Tietokantataulut on hajautettu useille MySQL-palvelimille. Jokainen solmu pitää tiedon keskusmuistissa mahdollistaen nopean toiminnan. NDB tukee hajautushakuja ja BTREE-indeksejä. (Sun Microsystems 2009e)

### 3.1.2 Indeksien toteutus

MySQL tallettaa rividatan ja indeksien datan eri paikkaan. Indeksointi ulottuu kaikkiin järjestelmän tukemiin tietotyyppihin. Lähteen (Sun Microsystems 2009e) mukaan paras tapa nopeuttaa SELECT-hakuoperaatioita on käyttää indeksejä haun kohteena olevissa sarakkeissa. MySQL tarjoaa tuen eri tallennusmoottoreille, jotka määrittelevät indeksien maksimipituuden ja maksimimäärän taulua kohti. Kaikki tallennusmoottorit tukevat ainakin 16 indeksiä taulua kohti, ja indeksirivin kokonaispituus voi olla kaikissa vähintään 256 tavua. Useimmilla tallennusmoottoreilla rajat ovat kuitenkin suuremmat. (Sun Microsystems 2009e)

Useimmat MySQL:n indeksityypit (perusavain (PRIMARY KEY), yksikäsitteinen indeksi (UNIQUE), B-puu-indeksi (INDEX) ja teksti-indeksi (FULLTEXT)) tallennetaan B-puu-tietorakenteeseen. Pistesiin, viivoihin ja monikulmioihin (spatial) liittyvät tietotyypit tallennetaan R-puihin, ja MEMORY-taulut tukevat myös hajautusindeksejä (hash index). Seuraavaksi esitellään tarkemmin nämä indeksityypit.

PRIMARY KEY (perusavain) on yksikäsitteinen indeksityyppi, jossa kaikkien avainkenttien tulee olla määriteltyjä, eivätkä ne osaa olla tyhjiä (määritelty attribuutilla NOT NULL). Taululla voi olla vain yksi PRIMARY KEY, joka voi olla myös monisarakkeinen. Mikäli jokin sovellus pyytää taulun PRIMARY KEY:tä, eikä sitä ole määritelty, MySQL palauttaa ensimmäisen yksikäsitteisen indeksin, jolla ei ole tyhjiä (null) arvoja perusavaimena. UNIQUE on yksikäsitteinen indeksityyppi, joka vaatii että indeksin jokaisen arvon tulee olla erilainen. (Sun Microsystems 2009e)

INDEX on MySQL:n indeksien oletustyyppi, joka tallennetaan B-puu-tietorakenteeseen. Mikäli indeksin luovassa käskyssä ei anneta muita määreitä, luodaan tämä oletusindeksi. (Sun Microsystems 2009e)

FULLTEXT on tekstin hakuun kehitetty indeksityyppi, jota voidaan käyttää vain MyISAM-tallennusmoottorin yhteydessä. FULLTEXT - indeksi voidaan luoda TEXT-, CHAR- tai VARCHAR - tyyppisille kentille tai näiden yhdistelmille. MySQL tukee kolmea eri hakutapaa FULLTEXT - indeksiin. Looginen haku (boolean search) tulkitsee hakusanaa käyttäen erikoisen kyselykielen sääntöjä. Oletuksena käytettävä luonnollisen kielen haku (natural language search) tulkitsee hakusanaa luonnollisen kielen lauseena. Kyselyn laajennushaku (query expansion search) on muunnelma luonnollisen kielen hausta, jossa ensin suoritetaan luonnollisen kielen haku. Tuloksesta poimitaan relevantimmat rivit, ja suoritetaan haku uudestaan niitä käyttäen. Haun tuloksena palautetaan toisen haun tulokset. (Sun Microsystems 2009e)

Hajautusindeksit voivat nopeuttaa kyselyitä erityisesti sellaisten taulujen kohdalla, jotka mahtuvat melkein kokonaan keskusmuistiin. InnoDB-tallennusmoottorin mekanismi valvoo taululle määritettyihin indekseihin kohdistuvia indeksihakuja. Jos mekanismi huomaa, että kyselyt hyötyisivät hajautusindeksistä, tallennusmoottori luo sellaisen automaattisesti. Hajautusindeksi rakentuu aina perustuen taulussa jo olevaan B-puu-indeksiin.

Hajautusindeksin ei tarvitse kuitenkaan kattaa koko B-puu-indeksiä, vaan se voi olla myös osittainen (partial). InnoDB-tallennusmoottori muodostaakin hajautusindeksit tarpeen mukaan niille indeksin sivuille, joihin viitataan usein. (Sun Microsystems 2009e)

### 3.2 Microsoft SQL Server

Microsoft SQL Server on Microsoftin kehittämä järeän luokan relaatiotietokantojen hallintajärjestelmä. Se ulottuu muutaman megatavun henkilökohtaisista tietokannoista aina monipalvelinympäristöissä toimiviin teratavujen kokoisii tietokantoihin. Tuotteena SQL Server on kuitenkin paljon muutakin kuin pelkkä tietokantamoottori. (Rankins ym. 2007)

SQL Serveristä on olemassa eri versioita riippuen ominaisuuksista ja käyttökohteista. Yleisimmät versiot ovat Standard Edition, Enterprise Edition ja Express Edition. Standard Edition -versioon kuuluu tietokantamoottorin lisäksi itsenäisiä palveluita. Enterprise Edition -versio on SQL Serverin täydellinen versio, joka sisältää tietokantamoottorin, lisäpalveluita ja joukon työkaluja SQL Server -klusterin rakentamiseen ja ylläpitämiseen. Lisäksi se Standard Edition -versiosta poiketen tukee useampia aktiivisia instansseja (klusterin solmuja) ja tarjoaa korkeamman käytettävyyden toimintoja kuten ajon aikana lisättävät muistit ja rinnakkaiset indeksit. Express Edition -versio on riisuttu ja ilmainen versio SQL Serveristä, joka sisältää myös tietokantamoottorin. Lisäksi se tukee vain yhtä prosessoria, 1 GB muistia ja 4 GB kokoisia tietokantoja. (Wikipedia 2009b)



### 3.2.1 Tietokantamoottori

Tietokantamoottori (Database engine, Relational engine) on toinen järjestelmän ydinkomponenteista, jota muut komponentit ja toiminnot edellyttävät. Se hyödyntää SQLOS<sup>1</sup>:n tarjoamia toimintoja toteuttaessaan relaatiotietovaraston (relational data store). Tietokantamoottori tarjoaa tyyppijärjestelmän, tallennusmoottorin sekä metodit datan nopeaan saatavuuteen. Sen tehtävänä on esimerkiksi optimoida SQL-kyselyt ja taata tietokannan turvallisuus. (Rankins ym. 2007)

Tietokantamoottori huolehtii kaikista tarvittavista tietorakenteista aina laitteistotasolta saakka taatakseen tiedon luotettavan säilytyksen. Rivit tallennetaan sivuihin (pages), joista jokainen on kooltaan 8 kilotavua. Sivut taas kootaan kahdeksan sivun kokoelmiin (extent).

SQL Server 2005 on SQL Server-tuoteperheen toiseksi uusin tuotantokäytössä oleva versio. Se julkaistiin lokakuussa 2005, ja se sisälsi natiivin tuen XML-muotoisen tiedon käsittelylle relaatiotiedon lisäksi. Tuoteperheen uusin versio SQL Server 2008 julkaistiin valmiina tuotantoon 6.8.2008. (Wikipedia 2009g)

### 3.2.2 Indeksien toteutus

Indeksit sisältävät avaimia, jotka on luotu yhdestä tai useammasta tietokantataulun tai -näkymän sarakkeesta. Nämä avaimet on tallennettu B-puu-tietorakenteeseen, jonka avulla SQL Server löytää rivin tai avaimeen viittaavat rivit nopeasti ja tehokkaasti. (Microsoft Corporation 2009)

SQL Server tukee seitsemää eri indeksityyppiä: ryvästetty indeksi (clustered index), ei-ryvästetty indeksi (nonclustered index), yksikäsitteinen indeksi

---

<sup>1</sup> SQLOS on rajapinta laitteiston/käyttöjärjestelmän ja SQL Serverin välillä.

(unique), indeksi johon sisältyy avaimettomia kenttiä (index with included columns), indeksoitu näkymä (indexed view), teksti-indeksi (full-text) sekä XML-indeksi. Seuraavaksi esitellään tuetut indeksityypit. (Microsoft Corporation 2009)

Ryvästetty indeksi käyttää tiedon järjestämiseen b-puu-rakennetta. Tieto tallentuu ryvästettyihin indekseihin avainarvojen perusteella. Yhdellä taululla voi olla vain yksi ryvästetty indeksi, sillä taulun rivit voidaan järjestää fyysisesti vain yhdellä tapaa. Vastaavasti ainoa tilanne, jolloin taulun rivit on tallennettu järjestyksessä, on silloin, kun taulu sisältää ryvästetyn indeksin. Kun taululla on ryvästetty indeksi, kutsutaan sitä ryvästetyksi tauluksi. Jos taululla ei ole ryvästettyä indeksiä, sen tietorivit tallennetaan järjestämättömään keko-rakenteeseen (heap). (Microsoft Corporation 2009)

Ei-ryvästetyt indeksit tallentuvat erilliseen tietorakenteeseen kuin taulun rivit. Ei-ryvästetty indeksi käyttää B-puu-rakennetta indeksin tallentamiseen. Jokainen avainarvo sisältää osoittimen taulun riviin, jolla kyseinen avainarvo sijaitsee. Tätä osoitinta kutsutaan rivin paikallistajaksi (row locator). Jos datarivit on tallennettu keko-rakenteeseen, rivin paikallistaja on osoitin tietoriville. Jos datarivit on tallennettu ryvästettyyn tauluun, rivin paikallistaja on ryvästetty indeksiavain (clustered index key). Indeksien rivit on tallennettu indeksiavaimen arvon mukaisessa järjestyksessä, mutta tietorivien ei taata olevan missään tietyssä järjestyksessä, ellei indeksiä ole luotu taululle. (Microsoft Corporation 2009)

Molemmat edellä esitellyt indeksityypit voivat olla yksikäsitteisiä (unique). Tämä tarkoittaa, että sama arvo ei voi toistua samassa taulussa. Muutoin kyseessä ei ole yksikäsitteinen indeksi. SQL Server ylläpitää automaattisesti indeksejä, kun tieto taulussa muuttuu. (Microsoft Corporation 2009)

Indeksi johon sisältyy avaimettomia kenttiä – tyyppin indeksi on laajennettu ei-ryvästetty indeksi, joka sisältää myös avaimettomia kenttiä avaimellisen

kenttien lisäksi. Tämä toisioindeksiä vastaava indeksityyppi mahdollistaa sellaisten ei-ryvästettyjen indeksien luomisen, jotka kattavat enemmän kyselyitä. Tämä on mahdollista siksi, että avaimettomat kentät voivat olla tietotyyppisiä, joita ei muuten sallita indeksin avainsarakkeiksi. Tietokantamoottori ei myöskään ota niitä huomioon laskiessaan indeksiavainsarakkeiden lukumääriä tai indeksiavainten kokoja. Suorituskyky voi olla huomattavasti parempi, sillä kyselyjen optimoija voi paikantaa kaikki sarakkeiden arvot suoraan indeksistä, eikä levyoperaatioita tietotauluun tai ryvästettyyn indeksitietoon tarvitse suorittaa. Avaimettomat sarakkeet tallennetaan vain lehtitasolla, kun yleensä sarakkeet tallennetaan kaikille indeksin tasoille. (Microsoft Corporation 2009)

Indeksoitu näkymä -tyypin indeksi on näkymä (view), jossa on yksikäsitteinen ryvästetty indeksi sekä lisäksi mahdollisesti yksi tai useampi ei-ryvästetty indeksi. Näkymä ei vaadi ylimääräistä tallennustilaa, koska tieto luetaan suoraan tietokantatauluista tarvittaessa. (Microsoft Corporation 2009)

Teksti-indeksin perusteet on kuvattu tarkemmin kohdassa 2.11 Teksti-indeksi. SQL Serverissä tämän toiminnallisen erikoisuuden tarjoaa MSFTESQL<sup>2</sup>. Teksti-indeksi on merkkiperustainen (token-based) indeksityyppi, jonka luontiprosessi poikkeaa useista muista indekseistä huomattavasti. B-puu-rakenteen sijaan teksti-indeksi tallennetaan käännettyyn (inverted), pinottuun (stacked) ja pakattuun (compressed) indeksirakenteeseen, joka perustuu indeksoitavan tekstin yksittäisiin merkkeihin (individual tokens). Full-text - indeksin koko SQL Server 2005:ssä on rajoitettu käytössä oleviin muistiresursseihin. (Microsoft Corporation 2009)

---

<sup>2</sup> MSFTESQL on lyhenne sanoista Microsoft Full-Text Engine for SQL Server. Kyseessä on nimensä mukaisesti tekstimoottori SQL Serveriä varten. (Microsoft Corporation 2009)

XML - indeksi on yleisnimitys indekseille, jotka koskevat xml-muotoista tietoa sisältäviä sarakkeita. XML-tieto tallennetaan XML-tyypin sarakkeisiin BLOB<sup>3</sup>-muodossa. Tällaiset binaarimuotoiset tietokokonaisuudet voivat olla jopa 2 GB kokoisia. Kun haku kohdistuu tähän tietojoukkoon, joudutaan tietokokonaisuudet ensin pilkkomaan (shredding), joka voi olla aikaa vievää. Luomalla indeksi kyseessä olevaan tietojoukkoon, voidaan hakuja nopeuttaa. On syytä kuitenkin huomata, että myös indeksin ylläpitäminen tiedon muuttuessa vaatii resursseja. XML - indeksejä on kahdenlaisia: ensisijaisia (primary) ja toissijaisia (secondary). Ensisijainen XML-indeksi on pilkottu esitys binaarimuotoisesta XML-tiedosta. Jokaista BLOB-tietuetta kohden indeksissä on useita rivejä tietoa. Jokainen rivi koostuu xml-tagin nimestä, xml-solmun arvosta, xml-solmun tyypistä (esim. attribuutti tai teksti), dokumentin järjestystiedosta, polusta xml-puun juureen sekä perustaulun perusavaimesta. Toissijaisia XML-indeksejä voidaan käyttää suorituskyvyn parantamiseen sellaisissa tauluissa, joissa on jo ensisijainen XML-indeksi. Toissijaisen XML-indeksin tyyppi voi olla joko PATH, VALUE tai PROPERTY. PATH-tyypin indeksiä käytetään eritoten silloin, kun kyselyissä käytetään polkuilmaisuja (path expression). Jos kyselyt pohjautuvat eri arvoihin eikä polkua ole määritelty tai polussa on käytetty jokerimerkkiä (wildcard), on VALUE-tyypin indeksi yleensä tehokkaampi. PROPERTY-tyypin indeksi on tarkoitettu kyselyihin, joissa vastauksena on yksi tai useampi arvo yksittäisestä XML-tiedosta. (Microsoft Corporation 2009)

SQL Server luo tiettyjä indeksejä automaattisesti käyttäjän näin halutessa. Jos käyttäjä esimerkiksi luo taulun ja määrittää sille perusavaimen PRIMARY KEY -määreellä, SQL Serverin tietokantamoottori luo automaattisesti indeksin tälle sarakkeelle. (Microsoft Corporation 2009)

---

<sup>3</sup> BLOB on lyhenne sanoista binary large object, binaarinen suuri objekti. Se on kokoelma binaarista tietoa, joka on tallennettu yhtenä kokonaisuutena tietokannan hallintajärjestelmään. (Elmasri & Navathe 2004)

### 3.3 Vertailu

Tuetut indeksityypit tuotteittain on esitelty taulukossa 1. MySQL:n osalta tiedot perustuvat lähteeseen (Sun Microsystems 2009e) ja Microsoft SQL Serverin osalta lähteeseen (Microsoft Corporation 2009). Indeksien nimen perusteella samankaltaisiksi päätellyt indeksityypit on alleviivattu. Tämän päätelmän pohjalta ei voida kuitenkaan todeta, etteikö näissä tietokantatuotteissa olisi myös muita samankaltaisia indeksityyppejä.

**Taulukko 1 Tietokantatuotteiden tukemat indeksityypit.**

MySQL	Microsoft SQL Server
<ul style="list-style-type: none"> <li>• perusindeksi</li> <li>• <u>yksikäsitteinen indeksi</u></li> <li>• B-puu-indeksi</li> <li>• <u>teksti-indeksi</u></li> <li>• hash-indeksit</li> </ul>	<ul style="list-style-type: none"> <li>• ryvästetty indeksi</li> <li>• ei-ryvästetty indeksi</li> <li>• <u>yksikäsitteinen indeksi</u></li> <li>• indeksi, johon sisältyy avaimettomia kenttiä</li> <li>• indeksoitu näkymä</li> <li>• <u>teksti-indeksi</u></li> <li>• xml-indeksi</li> </ul>

Tarkasteltaessa tuettujen indeksien määriä voidaan todeta, että Microsoft SQL Server tukee useampaa indeksityyppiä kuin MySQL. Taulukosta 1 voidaan nähdä, että Microsoft SQL Server tukee seitsemää indeksityyppiä ja MySQL viittä indeksityyppiä. Tämän toteamuksen pohjalta ei voida kuitenkaan tehdä johtopäätöksiä siitä, kumpi tietokantatuote tukee indeksien käyttöä laajemmin

tai puuttuuko niistä tuki jollekin olennaiselle indeksityypille. Seuraavaksi on esitetty eroavaisuuksia samankaltaisten indeksien toteutuksessa näiden tietokantatuotteiden osalta. Tarkastelu ja vertailu on tehty pohjautuen lähteisiin (Sun Microsystems 2009d) ja (Microsoft Corporation 2009).

Sekä MySQL- että Microsoft SQL Server-tietokantatuotteissa on tuki teksti-indeksityypille, joka on tarkoitettu sarakkeessa olevan tekstin hakuun kokonaisuudessaan tai osittain. Toteutuksessa ja käytössä on kuitenkin havaittavissa eroavaisuuksia. MySQL-tietokantatuotteen teksti-indeksityyppiä voidaan käyttää vain MyISAM-tallennusmoottorin yhteydessä ja sarakkeissa, joiden tyyppi on CHAR, VARCHAR tai TEXT. Indeksit tallennetaan B-puu-tietorakenteeseen. Indeksit voidaan määrittää CREATE TABLE -lauseella taulun luonnin yhteydessä, tai lisätä myöhemmin ALTER TABLE- tai CREATE INDEX -komennoilla. MySQL:n osalta ei lähteessä ole määritelty, kuinka monta teksti-indeksiä yhdessä tietokannan taulussa voi olla.

SQL Serverissä teksti-indeksin rakentamisen ja ylläpitämisen hoitaa MSFTESQL, joka tallentaa indeksin B-puusta poikkeavaan tietorakenteeseen. Yhdessä tietokannan taulussa tai indeksoidussa näkymässä voi olla vain yksi indeksi. Indeksit luodaan komennolla CREATE FULLTEXT INDEX.

Yksikäsitteisten indeksien käyttö on molemmissa tietokantatuotteissa hyvin samankaltainen. Tietokannan hallintajärjestelmä palauttaa virheen, mikäli tauluun yritetään lisätä uusi rivi käyttäen taulussa jo olevaa avainta. Tyhjiä arvojen käsittelyssä on kuitenkin havaittavissa eroavaisuuksia. SQL Serverissä tyhjiä arvoja kohdellaan tasavertaisena muiden arvojen kanssa, joten avainten osalta vain yhdellä rivillä voi olla tyhjä arvo. Lähteen mukaan on suositeltavaa valita NOT NULL -määritely sarake yksikäsitteiselle indeksille. MySQL:ssä on mahdollista käyttää tyhjiä arvoja kaikissa tallennusmoottoreissa lukuun ottamatta BDB:tä. Tyhjiä arvoja voidaan tallentaa sellaisiin sarakkeisiin, jotka voivat sisältää tyhjiä arvoja.

## 4 YHTEENVETO

Tietokannat kasvavat päivä päivältä yhä suuremmiksi. Siitä huolimatta vasteaika tiedon saamiseen käyttäjälle tulisi olla mahdollisimman pieni. Yhtenä ratkaisuna tähän ongelmaan voidaan pitää tietokannan indeksejä. Indeksi on tietokannan tauluun liitetty aputietorakenne, jonka avulla tietokannan hallintajärjestelmän on mahdollista hakea tiettyjä tietoja tietokannasta nopeammin. Indeksit voivat väärin käytettynä myös hidastaa tietokannan toimintaa, joten indeksin valitsemisessa ja suunnittelemisessa on oltava huolellinen.

Tässä kirjallisuuskatsaukseen perustuvassa tutkielmassa tarkasteltiin, minkä tyyppisiä indeksejä on olemassa ja miten niitä on hyödynnetty kahdessa suositussa tietokannan hallintajärjestelmässä, MySQL 5.0:ssa ja Microsoft SQL Server 2005:ssa. Tutkielman tarkoituksena oli selvittää, miten kahden valitun tietokantatuotteen indeksit poikkeavat toisistaan.

Eri indeksityypit on esitelty perustuen useaan eri lähteeseen. Lähteitä tutkittaessa voidaan todeta, että niiden tiedot ovat osaltaan ristiriitaisia esimerkiksi ryvästetyn indeksin kohdalla, josta käytetään eri nimityksiä lähteestä riippuen.

Tarkastelun kohteena olleet tietokantatuotteet on esitelty pintapuolisesti perustuen valmistajien antamiin tietoihin. Sen sijaan tarkemmin on tarkasteltu näiden tietokantatuotteiden indekseihin vaikuttavia seikkoja kuten tallennusmoottoreita sekä esitelty kunkin tuotteen tukemat indeksityypit ja niihin liittyvät erityispiirteet ja rajoitukset.

Vertailu näiden kahden tietokantatuotteen tukemien samantapaisten indeksien välillä on toteutettu perustuen valmistajien antamiin tietoihin. Vertailuun valittiin sellaiset indeksityypit, jotka nimensä perusteella ovat samantyyppisiä molemmissa tuotteissa. Tämän päätelmän pohjalta ei voida todeta, onko ko.

tuotteissa myös muita samantapaisia indeksityyppejä tai ovatko edes vertailuun valitut indeksityypit samantapaisia. Lisäksi tuettujen indeksien lukumäärän perusteella vertailu ei välttämättä kerro todellista kuvaa siitä, missä laajuudessa eri indeksityypit ovat tuettuina kussakin tietokantatuotteessa. Vertailun pohjalta voidaan nähdä kuitenkin keskeisiä eroja, mitä on havaittavissa indeksien toteutuksen osalta.

Eri indeksityyppien toteutusta ko. tietokantatuotteissa olisi voinut vertailla myös perustuen eri indeksi- ja tietorakenteisiin. Olisi esimerkiksi voitu verrata yksitasoisia ja monitasoisia indeksejä sekä tiiviisti ja harvasti toteutettuja indeksejä. Tässä tutkielmassa päädyttiin vain toteamaan eri rakenteiden olemassaolo ja kuvaamaan niiden toimintaperiaate.

Mahdollisia jatkotutkimuksen aiheita aihealueeseen liittyen voisi olla esimerkiksi selvittää, mikä indeksityyppi sopii parhaiten millekin tietojoukolle ja -tyypille, millaisia eroja suorituskyvyssä on eri indeksityyppien välillä ja miten indeksien tehokkuus on kehittynyt tietokantatuotteiden uusien versioiden myötä.



## LÄHDELUETTELO

- Connolly T. & Begg C. 2005. Database Systems: A Practical Approach to Design, Implementation, and Management. Harlow, England: Pearson Education Limited, 2005.
- Elmasri R. & Navathe S. 2004. Fundamentals of Database Systems, 4. painos. Addison-Wesley 2004.
- Gunderloy M. 2004. Understanding SQL Server Full-Text Indexing [online]. Saatavilla [www-osoitteesta](http://www.osoitteesta) <http://www.developer.com/db/article.php/3446891>. Haettu 9.3.2009.
- Guttman A. 1984. R-trees – A dynamic index structure for spatial searching [online]. Berkeley: University of California. Saatavilla [www-osoitteesta](http://www.osoitteesta) <http://www.sai.msu.su/~megeera/postgres/gist/papers/gutman-rtree.pdf>. Haettu 13.3.2009.
- Hovi A., Huotari J. & Lahdenmäki T. 2005. Tietokantojen suunnittelu & indeksointi, 2. laitos. Jyväskylä: Docendo 2005.
- Lightenna Limited. 2007. Database training [online]. Saatavilla [www-osoitteesta](http://www.osoitteesta) [http://www.lightenna.com/products/database\\_training\\_course](http://www.lightenna.com/products/database_training_course). Haettu 4.3.2009.
- Lightstone S., Teorey T. & Nadeau T. 2007. Physical Database Design. Elsevier Inc., 2007.
- Microsoft Corporation. 2009. SQL Server 2005 Documentation [online]. Saatavilla [www-osoitteesta](http://www.osoitteesta) <http://msdn.microsoft.com/en-us/library/ms203721%28SQL.90%29.aspx>. Haettu 13.3.2009.

- MySQL AB. 2004. MySQL: MySQL Storage Engine Architecture [online]. Saatavilla [www-osoitteesta http://dev.mysql.com/tech-resources/articles/storage-engine/part\\_1.html](http://dev.mysql.com/tech-resources/articles/storage-engine/part_1.html). Haettu 7.3.2009.
- Rankins R. 2007. Microsoft SQL Server 2005 Unleashed. Sams Publishing 2007.
- Sun Microsystems, Inc. 2009a. MySQL: Commercial License for OEMs, ISVs and VARs [online]. Saatavilla [www-osoitteesta http://www.mysql.com/about/legal/licensing/oem/](http://www.mysql.com/about/legal/licensing/oem/). Haettu 13.3.2009.
- Sun Microsystems, Inc. 2009b. MySQL: MySQL Customers [online]. Saatavilla [www-osoitteesta http://www.mysql.com/customers/?id=280](http://www.mysql.com/customers/?id=280). Haettu 7.3.2009.
- Sun Microsystems, Inc. 2009c. MySQL: NASA [online]. Saatavilla [www-osoitteesta http://www.mysql.com/news-and-events/generate-article.php?type=ss&id=nasa](http://www.mysql.com/news-and-events/generate-article.php?type=ss&id=nasa). Haettu 7.3.2009.
- Sun Microsystems, Inc. 2009d. MySQL: Sun to acquire MySQL [online]. Saatavilla [www-osoitteesta http://www.mysql.com/news-and-events/sun-to-acquire-mysql.html](http://www.mysql.com/news-and-events/sun-to-acquire-mysql.html). Haettu 7.3.2009.
- Sun Microsystems, Inc. 2009e. MySQL 5.0 Reference Manual [online]. Saatavilla [www-osoitteesta http://dev.mysql.com/doc/refman/5.0/en/](http://dev.mysql.com/doc/refman/5.0/en/). Haettu 7.3.2009.
- Sun Microsystems, Inc. 2009f. MySQL: Market Share [online]. Saatavilla [www-osoitteesta http://www.mysql.com/why-mysql/marketshare/](http://www.mysql.com/why-mysql/marketshare/). Haettu 13.3.2009.
- Wikipedia, 2009a. Klusteri (tietotekniikka) [online]. Saatavilla [www-osoitteesta http://fi.wikipedia.org/wiki/Klusteri\\_%28tietotekniikka%29](http://fi.wikipedia.org/wiki/Klusteri_%28tietotekniikka%29). Haettu 7.3.2009.

Wikipedia, 2009b. Microsoft SQL Server [online]. Saatavilla [www-osoitteesta http://en.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](http://en.wikipedia.org/wiki/Microsoft_SQL_Server). Haettu 5.3.2009.