Transaktionhallinta

- Transaktionhallinta (transaction management) on keskeinen tekijä tietokannan samanaikaisen käytön ja virheistä toipumisen kannalta.
- Useat prosessit voivat käsitellä tietokantaa samanaikaisesti
- Jos tietokoneessa on vain yksi prosessori, 'samanaikaiset' prosessit toimivat limittäin (interleaved) ja jakavat prosessorikapasiteettia. Jos koneessa on monta prosessoria, prosessit voivat toimia aidosti yhtä aikaa eri prosessoreilla (parallel). Jatkossa tarkastellaan limittäisyyteen perustuvaa samanaikaisuutta.

1

Transaktionhallinta

- Tietokantatransaktio on tietokantaa käsittelevä prosessin osa, jonka vaikutusten halutaan muodostavan yhden jakamattoman (atomisen) kokonaisuuden.
  - voi sisältää hakuja, lisäyksiä, poistoja muutoksia
  - Esim. pankkitilisovelluksen proseduuri tilisiirto(t1, t2, x), joka siirtää x mk tililtä t1 tilille t2:

```
begin transaction
  update tili set saldo=saldo-x where tilinro=t1;
  update tili set saldo=saldo+x where tilinro=t2;
  insert into tilitapahtumat values
    (pvm, time, 'siirto', x, t1, t2, ...);
  commit;
end transaction;
```

2

Transaktionhallinta

- Esimerkissä jakamattomuus merkitsee sitä, että kaikki 3 tietokantaoperaatiota suoritetaan eikä vain osaa niistä
- Miksei sitten suoritettaisi?
  - Käsittelyssä voi sattua häiriöitä missä vaiheessa tahansa. Nämä voivat johtua ulkoisista syistä tai olla tkhj:n itse aiheuttamia, jotta se voisi jatkaa toimintaansa (esim. syntyy lukkiutuma, joka pitää purkaa)
  - Voisi siis käydä siten, että tilin t1 sivu on kirjoitettu levylle ja tilin t2 sivu jää kirjoittamatta – tällöin operaatio ei olisi jakamaton

3

Transaktionhallinta

- Tietokantatransaktioilta edellytetään 4 perusominaisuutta (ACID vaatimukset):
- Atomisuus (**a**tomicity)= jakamattomuus:
  - kaikki transaktion tietokantamuutokset suoritetaan tai ei mitään niistä.
- Eheyden säilyttäminen (**c**onsistency):
  - transaktio siirtää tietokannan ehyestä tilasta toiseen ehyeen tilaan. Ehyt tila on tila, jossa tietokantaan liittyvät eheysehdot ovat voimassa. Tämän vaatimuksen toteutuminen on käyttäjän vastuulla.

4

Transaktionhallinta

- Eristyvyys (**i**solation)
  - Muut samanaikaiset transaktiot eivät sotke transaktion suoritusta. Transaktio suoritetaan ikään kuin muita samanaikaisia transaktioita ei olisikaan. Transaktion kannalta näyttää siltä, että kaikki muut transaktiot on suoritettu joko ennen sitä tai sen jälkeen.
- Pysyvyys (**d**urability)
  - Kun tkhj ilmoittaa käyttäjälle, että transaktio on menestyksekkäästi päätetty, sen tekemät muutokset tietokantaan jäävät voimaan (kunnes jokin toinen transaktio ne kumoaa). Mikään häiriö ei niitä hävitä.

5

Transaktionhallinta

Tarkastellaan tilisiirto (1234, 5678, 5000) tapahtuman operaatiota:

1. transaktion aloitus(pyyntö) begin
2. lukuoperaatioita tietohakemiston sivuihin
3. lukuoperaatioita tili-taulun tilinumeropohjaiseen indeksiin
4. luetaan puskuriin b se tili-taulun sivu p, jossa on tilin 1234 rivi

   --- ohjelmallisesti varsinainen tililtäotto ja siihen mahdollisesti liittyvät tarkistukset + lisätoimet ---
5. kirjoitetaan muutetun rivin sisältö sivulle p
6. -9. vastaavat operaatiot tilin 5678 sivulle (tililepano)
10. lukuoperaatioita tietohakemistoon
11. luetaan tilitapahtuma-taulun viimeinen sivu (oletetaan, että rakenne on kasa)
12. lisätään tilitapahtuma tietue tilitapahtumasivulle
13. transaktion sitoutumispyyntö (commit).

6

Transaktionhallinta

- Yleisesti kannan käsitttely muodostuu luku- ja kirjoitusoperaatioista
- read(X,v)
  - lukee tietoalkion X muuttujaan v
- write(X,v)
  - kirjoita tietoalkio X muuttujasta v

7

Transaktionhallinta

- read(X,v):
  - Selvitä X:n sisältävän sivun osoite p
  - Pyydä puskurienhallintaa lataamaan sivu p (lukee sivun, jos se ei ole jo jossain puskurissa). Puskurienhallinta naulitsee sivun puskuriin (fix) ja palauttaa puskurin osoitteen b
  - Kopioi tietoalkio X puskurista b muuttujaan v
  - Ilmoita puskurienhallinnalle sivun vapautuksesta (unfix)

8

Transaktionhallinta

- write(X,v)
  - Selvitä X:n sisältävän sivun osoite p
  - Pyydä puskurienhallintaa lataamaan sivu p muutosta varten (lukee sivun, jos se ei ole jo puskurissa). Puskurienhallinta naulitsee sivun puskuriin (fix) ja palauttaa puskurin osoitteen b
  - Kopioi tietoalkio X muuttujasta v puskuriin b
  - Ilmoita puskurienhallinnalle sivun muuttamisesta ja vapautuksesta (unfix)
  - Puskurienhallinta kirjoittaa aikanaan muuttuneet sivut levylle.
- Sekä lukuun että kirjoitukseen liittyy yllä kuvatun lisäksi samanaikaisuuden hallintaan liittyviä toimia (lukituksia)

9

Transaktionhallinta

- Transaktion sisäisiä vaiheita hallitaan määrittelemällä transaktion tilasiirtymämalli ja seuraavat tilat:

1. alkutila: transaktio syntyy (generoidaan)
   - transaktiolle annetaan oma tunniste, jne.
2. aktiivinen: transaktio suorittaa varsinaisia operaatioitaan (read(), write())
3. osittain sitoutunut (partially committed): transaktion ohjelmakoodi on suoritettu ja se on pyytänyt sitoutumista (commit-operaatiolla)

10

Transaktionhallinta

4. sitoutunut (committed): tkhj on vahvistanut transaktion tietokantaan tekemät muutokset pysyviksi eli sitoutuminen on onnistunut
   - Tietokannan muutokset eivät ole enää peruttavissa (ilman uutta transaktiota). Tiedot eivät kuitenkaan ole välttämättä levyllä asti (vahvistus on vain 'looginen'), kuitenkin vastaava lokitieto on tässä vaiheessa levyllä

11

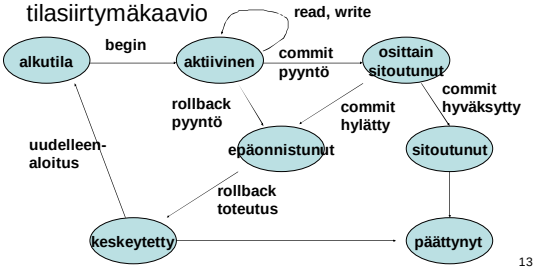
Transaktionhallinta

5. epäonnistunut (failed): sitoutuminen on epäonnistunut (esim. samanaikaisuuden hallintaan liittyvien tarkistusten takia) tai transaktio itse on suorittanut keskeytysoperaation rollback (abort)
6. keskeytetty: epäonnistunut transaktio on peruutettu eli tietokanta on palautettu ennen transaktion aloitusta vallinneeseen tilaan
7. päättynyt (terminated): transaktion olemassaolo lakkaa

12

Transaktionhallinta

- Keskeytetty transaktio voidaan aloittaa uudelleen päättynyttä ei voida. Alla tilasiirtymäkaavio



13

Transaktionhallinta

- SQL:ssä transaktio päätetään normaalisti commit-lauseella. Samalla commit aloittaa uuden transaktion. Joissain ympäristöissä (esim. sulautettu SQL) voi käyttää myös begin transaction ja end transaction lauseita.
- Transaktion voi lopettaa myös rollback-lauseella.
- Rollback peruuttaa pääsääntöisesti koko transaktion.
  - Osittainen peruutus saatavissa aikaan jatkoaloituskohdilla (savepoint)

14

Transaktionhallinta

```
commit;
muutoksia 1
/* tallenna tilanne, nimeä kohta = p1 */
savepoint p1;
muutoksia 2
/*peruuta transaktiossa pisteeseen p1, muutoksia
   1 säilyy */
 rollback to savepoint p1;
```

15

Transaktionhallinta

- Samanaikaisten toimintojen salliminen ja häiriöiden vaikutusten eliminointi muodostavat monimutkaisen toiminnallisen kokonaisuuden.
- 'Pelisääntönä' on karkeasti se, että samanaikaisuutta rajoitetaan vain niin paljon, ettei 'normaalitilanteessa' jouduta usein korjaamaan samanaikaisuudesta aiheutuneita (tulossa olevia) vaurioita.
- Rajoituskeinoja:
  - lukitaan tietoalkioita muilta lukoilla (lock);
    - pitkä lukinta-aika rajoittaa hankalasti muiden transaktioiden etenemistä; voi syntyä lukkiutuma, jolloin mikään transaktio ei pääse etenemään
  - optimistiset menetelmät: annetaan mennä; tarkistetaan; korjataan, jos tarpeen

16

Transaktionhallinta

- Transaktioloki (transaction log):
  - Transaktioloki on keskeinen transaktionhallinnan väline
  - Kaikki transaktioiden suorittamat operaatiot kirjataan lokiin, johon mm. häiriöiden korjaus (tietokannan elvytys) perustuu.

17

Transaktionhallinta

- Loki (log) on peräkkäistiedosto, jonne viedään
  - tapahtuman aloituskirjauksia (start,T)
  - muutoskirjauksia (write,T,x,v,u):
    - tapahtuma T on muuttanut tietoalkion x vanhan arvon v (alkukuva, before image) uudeksi arvoksi u (jälkikuva, after image)
  - sitoutumiskirjauksia (commit,T): tapahtumaan T on sitouduttu
  - keskeytyskirjauksia (abort,T): tapahtuma T on peruttu
  - tarkistuspisteitä (checkpoint)
  - [lukukirjauksia (read,T,x) – ei tarvita elvytyksessä]

18

Transaktionhallinta

- Jokaisella lokikirjauksella on yksikäsitteinen tunnus (log sequence number, LSN)
- Lokin muutoskirjauksessa esiintyvä tietoalkio x voi olla periaatteessa mitä tahansa yksittäisestä merkistä koko sivuun. Tyypillisiä:
  - Sivu
    - loki kasvaa suureksi
  - Relaation rivi
    - lokitiedosto vie vähemmän tilaa kuin kokonaisten muutettujen sivujen kirjaus.
  - Rivin tunnustiedot +tietoalkion tunnustiedot + tietoalkion arvo
    - vielä vähän tilaa, mutta käsittely työläämpää

19

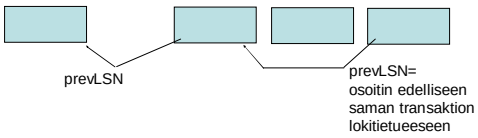
Transaktionhallinta

- Lokitiedostoa käsitellään kuten muitakin tiedostoja, ts. lokitietue viedään ensin puskurisivulle, sitten aikanaan esim. puskurisivun täyttyessä tai viimeistään transaktion sitoutuessa levylle.
  - lokitietueet voidaan myös 'pakkokirjoittaa' levylle (kirjoitetaan välittömästi)
- Häiriön sattuessa vain levyllä oleva lokin osa on varmuudella käytettävissä.
  - Yleensä lokin käytettävyyttä varmistetaan vielä levyvirheiden varalta esimerkiksi useamman tiedoston vuorottelulla, ja varmistamalla levyllä oleva loki ajoittain nauhalle (tai toiselle levy-yksikölle)).

20

Transaktionhallinta

- Samaan tapahtumaan liittyvät lokitietueet voidaan kytkeä ketjuksi:



21

Transaktionhallinta

- Puskurien sisällön viennissä levylle on erilaisia periaatteita:
  - välitön päivitys (immediate update): sivut viedään levylle ennen niitä muuttaneiden tapahtumien sitoutumista
  - viivästetty päivitys (deferred update): sivut viedään levylle vasta niitä muuttaneiden tapahtumien sitoutumisen jälkeen

---

  - pakotettu (forced): tehdään heti
  - vapaasti (no-force): tehdään viimeistään kun puskuritilaa tarvitaan muuhun käyttöön. Muut transaktiot voivat käyttää puskurissa olevaa muuttunutta tietoa, ilman levyhakuja

22

Transaktionhallinta

- Kirjoitus puskurista levylle voi olla tarpeen puskurin vapauttamiseksi muuhun käyttöön
- Välitön päivitys voi sallia puskurin varastamisen (frame stealing), eli sivu viedään puskurista levylle ennen kuin muutokseen on sitouduttu ja puskuri annetaan toisen sivun käyttöön.
- Levylle viety data on likaista (dirty), muuttuen 'puhtaaksi' vasta kun muutoksen aiheuttaneeseen tapahtumaan on sitouduttu. Likaisen datankin luku voidaan kontrolloidusti sallia.

23

Transaktionhallinta

- Commit-pyynnön toteuttaminen aiheuttaa tkhj:ssa joukon samanaikaisiin transaktioihin liittyviä tarkistuksia.
- Sitoutuminen tehdään ns. sitoutumiskäytännön (commit protocol) mukaisesti:
  - kirjoitetaan lokiin sitoutumismerkintä (commit, T)
  - pakkokirjoitetaan loki levylle (kirjoitetaan levylle kaikki lokin sivut, jotka ovat toistaiseksi vasta puskureissa)
  - vapautetaan transaktion T mahdollisesti varaamat resurssit (mm. samanaikaisia toimintoja säätelevät lukot transaktion käsittelemiin tietoalkioihin).
  - kuitataan sitoutuminen tehdyksi
- Huom. Lokitietue viedään levylle aina ennen dataa. Tätä sanotaan WAL-käytännöksi (write-ahead-logging), ja sitä noudatetaan myös kirjoitettaessa tietosivuja levylle välittömän päivityksen menetelmällä.

24

Transaktionhallinta

- Rollback-pyynnön toteutus aiheuttaa tapahtuman tekemien muutosten peruutuksen
  - Käydään läpi lokista löytyvät peruttavaan transaktioon liittyvät kirjoitusmerkinnät lopusta alkuun (siis ketjua pitkin) ja korvataan kukin tietoalkio alkukuvallaan.
  - Kirjoitetaan lokiin keskeytysmerkintä (abort)
- Jos jokin toinen transaktio olisi lukenut sellaisen tietoalkion arvon, joka peruutuksessa korvautuu vanhalla arvolla, pitäisi myös tämä toinen transaktio perua (ns. cascading rollback, vyöryvä peruutus)
  - käytännössä näin ei kuitenkaan käy, sillä tapahtumille hyväksytään vain sellaisia suoritusjärjestyksiä, joissa tämä vältetään.

25

Transaktionhallinta

Tarkistuspiste (checkpoint)
- Tarkistuspisteessä viedään levylle asti kaikki puskureissa olevat tietokantasivujen päivitykset

Tarkistuspiste sisältää seuraavat toiminnot:
1. estetään väliaikaisesti transaktioiden suoritus
2. pakkokirjoitetaan kaikki transaktioiden päivittämät sivut puskurista levylle
3. kirjoitetaan lokiin tarkistuspistekirjaus ja pakkokirjoitetaan loki levylle
4. sallitaan transaktioiden jatkaa suoritustaan

26

Transaktionhallinta

- Transaktioiden estäminen ja vaiheen 2 kesto voivat hidastaa normaalitoimintaa.
- Vaiheiden 2 ja (3, 4) järjestys voidaan vaihtaa, kun säilytetään edellinen loppuun suoritettu tarkistuspiste (viimeisenä virallisena), kunnes kaikki sivut ovat levyllä.
- Tarkistuspisteiden taajuus on tietokannan hoitajan päätettävissä.
  - Järkevä taajuus riippuu tietokannan päivitystiheydestä ja häiriöalttiudesta:
    - esim. 4 kertaa tunnissa tai
    - kun tietty määrä transaktioita on sitoutunut edellisen tarkistuspisteen jälkeen.

27

Transaktionhallinta - elvytys

- Transaktion jakautumattomuus tai pysyvyys voi vaarantua monen häiriötilanteen takia:
  - Tietokonejärjestelmä ‘romahtaa’ (system crash) laitteisto-, ohjelmisto- tai tietoliikennevirheen takia. Yleensä keskusmuistin (tietokantapuskurien) sisältöä menetetään.
  - Yksittäisen transaktion suoritus keskeytyy ohjelman poikkeustilanteen (nollalla jako tms.) tai loogisen ohjelmavirheen takia. Käyttäjä voi myös keskeyttää kyselyn suorituksen ‘väkivalloin’.

28

Transaktionhallinta - elvytys

- Transaktion suoritus keskeytetään hallitusti esim. transaktion (proseduurin) koodissa suoritetaan jonkin ehdon seurauksena rollback-pyyntö.
  - Jos ei esimerkiksi ei löydy transaktion tarvitsemaa syötettä tai se on virheellinen.
- Samanaikaisuuden hallinnan alijärjestelmä joutuu keskeyttämään transaktion, jotta muut transaktiot voisivat edetä (lukkiutuma tai jokin lievempi suoritusjärjestykseen liittyvä häiriö).

29

Transaktionhallinta - elvytys

- Levyvirhe on turmellut levyn sisältöä. (harvinaista)
- Ulkopuolinen häiriötekijä (operointivirhe, … , sähkökatko) keskeyttää transaktion. (harvinaista)
- Yllä olevien kohdalla elvytys voi sisältää edellisen varmuuskopion (ajanhetkeltä t) käyttöönoton
- Lokin avulla voidaan mahdollisesti suorittaa uudelleen (redo) hetken t ja häiriöajankohdan välillä suoritetut toiminnot
  - Varmuuskopion ja lokin tulisi olla esim. nauhalla tallessa (levyvirhe …).

30

Transaktionhallinta - elvytys

- Lokiin perustuvan elvytyksen periaatteet:
  - peruutetaan (undo) ne muutokset, joita keskeytyneet transaktiot ovat tehneet levyille
  - suoritetaan uudelleen (redo) sellaisten sitoutuneiden transaktioiden suorittamat tietokantapäivitykset, joita ei häiriön sattuessa ollut ehditty kirjoittaa levylle (vaan vasta puskurissa olevaan sivuun)
- Normaalitilanteessa undo- ja redo-toimet kohdistuvat siihen tietokannan tilaan, joka oli häiriön sattuessa voimassa (ei aikaisempaan varmuuskopioon). Lokin (ja mahdollisesti muiden tietojen) avulla etsitään, mitä pitää tehdä.
- Varmuuskopioita tarvitaan aika harvoin ...

31

Transaktionhallinta - elvytys

- Elvytys voi tapahtua eri tavoin riippuen siitä, mitä on tehty valmiiksi häiriön sattuessa
  - ei ole tehty mitään
  - tieto viety levylle
  - muutos kirjattu vain puskuriin

32

Transaktionhallinta - elvytys

Loki sitoutumisessa:

- Transaktio T on sitoutunut silloin ja vain silloin, kun lokista löytyy sille sitoutumismerkintä (commit, T)
- Levyllä olevien lokitietueiden perusteella selviää,
  - mitkä transaktiot olivat häiriön sattumishetkellä sitoutuneita ja mitkä kesken
  - mitä on kirjoitettu (+ vanha arvo): write-tietue

33

Transaktionhallinta - elvytys

Yleinen lokiin perustuva elvytysalgoritmi:

1. Luetaan lokia levyltä ja muodostetaan kaksi transaktiolistaa:

- Keskeneräiset = transaktiot, joille on lokissa aloituskirjaus (start), mutta ei sitoutumiskirjausta (commit) eikä viimeistä tarkistuspistettä edeltävää keskeytyskirjausta (abort)
  - keskeytyskirjaus ennen tarkistuspistettä:
    - muutosten peruutukset on merkitty puskuriin ja huomattu tarkistuspisteessä
- Sitoutuneet = transaktiot, joille on lokissa sitoutumiskirjaus viimeisen tarkistuspisteen jälkeen

34

Transaktionhallinta - elvytys

2. Perutaan Keskeneräiset-listan transaktioiden kirjoitus-operaatiot selaamalla lokia lopusta alkuun päin:
   - jokaista löytyvää muutoskirjausta (write,T,x,vanha,uusi) kohti suoritetaan operaatio write(x, vanha) (palautetaan tietoalkion x alkukuva voimaan)
3. Uusitaan Sitoutuneet-listan transaktioiden kirjoitus-operaatiot selaamalla lokia alusta loppuun päin:
   - jokaista löytyvää muutoskirjausta (write,T,x,vanha,uusi) kohti suoritetaan operaatio write(x, uusi) (saatetaan siis tietoalkion jälkikuva uudelleen voimaan)

35

Transaktionhallinta - elvytys

- Esimerkki. Olkoon levyllä oleva lokin sisältö häiriötilanteessa seuraava

1: (start, T1) 2: (start, T2) 3: (write, T1, x1, 'AAA', 'BBB') 4: (commit, T1) 5: (write, T2, X1, 'BBB', 'CCC') 6: (checkpoint) 7: (write, T2, x2, '0000', '1111') 8: (start, T3) 9: (commit, T2) 10: (write, T3, x1, 'CCC', 'DDD') 11: (write, T3, x2, '1111', '2222')	Elvytysalijärjestelmä lukee lokia levyltä ja muodostaa listat Kesk = <T3>, Sit = <T2> peruu transaktion T3 operaatiot suorittamalla: write(x2, '1111') (11) write(x1, 'CCC') (10) suorittaa uudelleen transaktion T2 operaatiot write(x1, 'CCC') (5) write(x2, '1111') (7)

36

Transaktionhallinta - elvytys

- Edellä kuvatussa algoritmissa tehdään uusinnan yhteydessä turhia kirjoituksia:
  - ennen tarkistuspistettä suoritettua kirjoitusoperaatiota ei tarvitse uusia (esimerkissä rivi 5), sillä se on tarkistuspisteessä hoidettu levylle.
  - tarkistuspisteen jälkeen tehtyä kirjoitusoperaatiota, jonka tulos on ehtinyt levylle ennen häiriötilannetta, ei tarvitse uusia
    - Tilanne saadaan selville tallentamalla datasivun tunnustietueeseen viimeisen sivulle tehtyä päivitystä vastaavan lokitietueen tunnus (kenttä pageLSN)
  - Myös perumiskirjoitus (undo) voi olla turha, sillä sellaisia kirjoituksia, joiden tulos ei ole ehtinyt levylle, ei tarvitse perua
- uusimis-/perumiskirjoituksen tarve selviää vertaamalla lokitietueen tunnusta (LSN) ja datasivulla olevaan viimeisimmän päivittäjän tunnusta (pageLSN –kenttää):
  - undo tarvitaan, jos LSN <= pageLSN (eli jos muutos on ehtinyt levylle)
  - redo tarvitaan, jos LSN > pageLSN (eli muutos ei ole vielä ehtinyt levylle)

37

Transaktionhallinta - elvytys

- Jos pageLSN(x1) = 10, pageLSN(x2) = 7, niin vain write(x1,'CCC') tarvitaan

1: (start, T1)
2: (start, T2)
3: (write, T1, x1, 'AAA', 'BBB')
4: (commit, T1)
5: (write, T2, X1, 'BBB', 'CCC')
6: (checkpoint)
7: (write, T2, x2, '0000', '1111')
8: (start, T3)
9: (commit, T2)
10: (write, T3, x1, 'CCC', 'DDD')
11: (write, T3, x2, '1111', '2222')

Elvytysalijärjestelmä lukee lokia levyltä ja muodostaa listat
Kesk = <T3>, Sit = <T2>
peruu transaktion T3 operaatiot suorittamalla:
~~{write(x2, '1111') (l11<=7)}~~
write(x1, 'CCC') (10=10)
suorittaa uudelleen transaktion T2 operaatiot
~~{write(x1, 'CCC') (l5>10)}~~
~~{write(x2, '1111') (l7>7)}~~

38

Transaktionhallinta - elvytys

- Häiriö voi sattua myös kesken elvytyksen
  - Tällöin oleellista on, että elvytysprosessin peräkkäiset suoritukset tuottavat saman tuloksen
- Edellä kuvattu elvytys ei kirjaa lokiin mitään, joten siinä lähdetään uudelleen liikkeelle samasta lähtötilanteesta, pageLSN kentät sivuilla ovat voineet muuttua, jos sivu on viety levylle, mutta lopputulos on sama.
- Käytännössä kirjataan peruutusten yhteydessä lokiin yleensä myös kompesaatiomerkintöjä, joista näkee mitä peruutuksia on tehty. Näitä ei kuitenkaan tarvita elvytyksessä.
- Elvytyksestä on erilaisia muunnelmia riippuen siitä minkälaista puskurien levykirjoituspolitiikkaa käytetään.

39