



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 (parallele). 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. pankkiliisovelluksen 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 (syntyy lukkiutuma, joka pitää purkaa)
 - Voisi siis käydä siten, että tilin t1 sivu on kirjoitettu levyille ja tilin t2 sivu jää kirjoittamatta – tällöin operaatio ei olisi jakamaton

3



Transaktionhallinta

- Tietokantatransaktioilta edellytetään 4 perusominaisuutta (**ACID vaatimukset**):
- **Atomisuus (atomicity)**:
 - kaikki transaktion tietokantamuutokset suoritetaan tai ei mitään niistä.
- **Eheyden säilyttäminen (consistency)**:
 - transaktio siirtää tietokannan eheydestä tilasta toiseen ehyeen tilaan. **Ehyt tila on tila, jossa tietokantaan liittyvät eheyshdot ovat voimassa**. Tämän vaatimuksen toteutuminen on käyttäjän vastuulla.

4



Transaktionhallinta

- **Eristyvyys (isolation)**
 - 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 (durability)**
 - 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) operaatiota:

1. transaktion aloitus(pyynnö) begin
2. lukuoperaatioita tietohakemiston sivuihin
3. lukuoperaatioita tili-relaation hakemistoon
4. luetaan puskuriin b se tili-relaation sivu p, jossa on monikko 1234
 - ohjelmallisesti varsinainen liittäotto ja siihen mahdollisesti liittyvät tarkistukset + lisätoimet ---
5. kirjoitetaan muutetun monikon sisältö sivulle p
- 6.-9. vastaavat operaatiot tilin 5678 sivulle (tilillepano)
10. lukuoperaatioita tietohakemistoon
11. luetaan tilitapahtuma-relaation viimeinen sivu (oletusrakenne kasa)
12. lisätään tilitapahtuma tietue tilitapahtumasivulle
13. transaktion sitoutumispyyntö (commit).

6



Transaktionhallinta

- Yleisesti kannan käsittely muodostuu luku- ja kirjoitusoperaatioista
- **read(X,v)**
 - lukee tietokalkion X muuttujaan v
- **write(X,v)**
 - kirjoita tietokalkio X muuttujasta v
- E&N: käyttää muotoa (**read_item(X)**, ja **write_item(X)**)

7



Transaktionhallinta

- **read(X,v):**
 - Selvitä X:n sisältävän sivun osoite p
 - Pyydä puskurienhallintaa lataamaan sivu p (lukee, jos sivu ei ole jo puskurissa). Puskurienhallinta nautitsee sivun puskuriin (fix) ja palauttaa puskurin osoitteen b
 - Kopioi tietokalkio 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, jos sivu ei ole jo puskurissa). Puskurienhallinta nautitsee sivun puskuriin (fix) ja palauttaa puskurin osoitteen b
 - Kopioi tietokalkio X muuttujasta v puskuriin b
 - Ilmoita puskurienhallinnalle sivun muuttamisesta ja vapautuksesta (unfix)
 - Puskurienhallinta kirjoittaa aikanaan muuttuneet sivut levyille.
- 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)
 - 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): tkh:n 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 vain 'looginen'), mutta vastaava **lokietieto** on levyllä

11



Transaktionhallinta

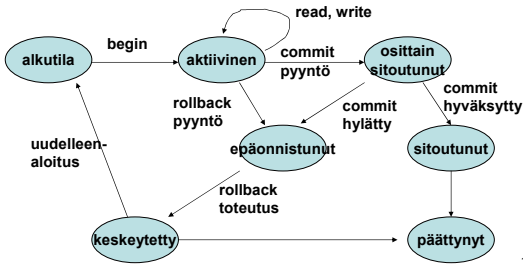
5. **epäonnistunut** (failed): sitoutuminen on epäonnistunut (esim. samanaikaisuuden hallintaan liittyvien tarkistusten takia) tai transaktio itse on suorittanut keskeytysoperaation rollback (abort)
6. **keskeytetty**: epäonnistunut transaktio on peruutettu eli tietokanta on palautettu ennen transaktion aloitusta vailinneeseen 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 sen aiheuttamia (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
 - optimistisia menetelmiä: annetaan mennä; tarkistetaan; korjataan, jos tarpeen

16



Transaktionhallinta

- Transaktioiden T_1, \dots, T_n **ajojärjestys (schedule, history)** on näiden operaatioista (luku/kirjoitus) muodostuva jono, jossa kunkin transaktion T_i operaatiot ovat samassa järjestyksessä kuin ne ovat transaktion T_i sisällä. Muiden transaktioiden operaatioita voidaan kuitenkin suorittaa T_i :n operaatioiden välissä
 - $T_1 = [t_{11}, t_{12}, t_{13}]$, $T_2 = [t_{21}, t_{22}]$, $T_3 = [t_{31}, t_{32}, t_{33}]$
 - $S = [t_{11} \ t_{31} \ t_{12} \ t_{21} \ t_{22} \ t_{32} \ t_{13} \ t_{33}]$
- Transaktioiden todella suorittamat operaatiot kirjataan **lokiin (log)**, johon mm. häiriöiden korjaus (tietokannan elvytys) perustuu.

17



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 (hollallajako tms.) tai loogisen ohjelmavirheen takia. Käyttäjä voi myös keskeyttää kyselyn suorituksen 'väkivalloin'.

18



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).

19



Transaktionhallinta - elvytys

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

20



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 levyille (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 ...

21



Transaktionhallinta - elvytys

- Elvytys voi tapahtua eri tavoin riippuen siitä mitä on tehty valmiiksi häiriön sattuessa
 - ei ole tehty mitään
 - tieto viety levyille
 - muutos kirjattu vain puskurin
- Puskurien sisällön viennissä levyille on erilaisia periaatteita:
 - **välitön päivitys** (immediate update): viedään levyille ennen sitoutumista
 - **viivästetty päivitys** (deferred update): viedään levyille sitoutumisen jälkeen
 - pakotettu (forced): tehdään heti
 - vapaasti (no-force): viimeistään kun puskuritilaa tarvitaan muuhun käyttöön. Muut transaktiot voivat käyttää puskurissa olevaa muuttunutta tietoa, ilman levyhakuja

22



Transaktionhallinta - elvytys

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

23



Transaktionhallinta - elvytys

- Loki (log) on peräkkäistiedosto, jonne viedään
 - tapahtuman aloituskirjauksia (start,T)
 - muutoskirjauksia (write,T,x,v1,v2):
 - tapahtuma T on muuttanut tietoalkion x vanhan arvon v1 (alkukuva, before image) uudeksi arvoksi v2 (jälkikuva, after image)
 - sitoutumiskirjauksia (commit,T): tkhj on sitoutunut
 - keskeytyskirjauksia (abort,T): tkhj on peruuttanut
 - tarkistuspisteitä (checkpoint)
 - [lukurjauksia (read,T,x) – ei tarvita elvytyksessä]

24