

# Samtidighetsfenomener og anomalier i eksekveringsplaner (kursorisk)

# Liste over fenomener

- P0 – Skitten skriv  $w_1(x)..w_2(x)..(c_1 \text{ eller } a_1)$
- P1 – Skitten les  $w_1(x)..r_2(x)..(c_1 \text{ eller } a_1)$
- P2 – Ikke-repeterbar les  $r_1(x)..w_2(x)..(c_1 \text{ eller } a_1)$
- P3 – Fantomfenomen  $r_1(Q)..w_2(y \text{ i } Q)..(c_1 \text{ eller } a_1)$
- P4 – Tapt oppdatering  $r_1(x)..w_2(x)..w_1(x)..c_1$

I P3 står  $Q$  for et predikat, dvs. svaret på et where-uttrykk.  $w_2(y \text{ i } Q)$  betyr at skriveoperasjonen  $w_2(y)$  kan øke resultatmengden til  $Q$ .

# Liste over anomalier

- A3A – Fantomleseanomali

$$r_1(Q) \dots w_2(y \text{ i } Q) \dots c_2 \dots r_1(Q) \dots c_1$$

- A3B – Fantomskjevskrivning

$$r_1(Q) \dots r_2(Q) \dots w_1(y \text{ i } Q) \dots w_2(z \text{ i } Q) \dots (c_1 \text{ og } c_2)$$

- A5A – Skjevlesing

$$r_1(x) \dots w_2(x) \dots w_2(y) \dots c_2 \dots r_1(y) \dots c_1$$

- A5B – Skjevskrivning

$$r_1(x) \dots r_2(y) \dots w_1(y) \dots w_2(x) \dots (c_1 \text{ og } c_2)$$

- A6 – Lesetransaksjonsanomali

$$r_2(x) \dots r_2(y) \dots w_1(y) \dots c_1 \dots r_3(x) \dots r_3(y) \dots c_3 \dots w_2(x) \dots c_2$$

# Skitten skriv

$T_1$  oppdaterer  $x$ . Deretter oppdaterer  $T_2$   $x$  før  $T_1$  committer eller avbryter. Hvis  $T_1$  eller  $T_2$  forsøker å gjøre en rollback, er det uklart hva som er korrekt verdi:  $T_2$ s endringer forsvinner dersom  $T_1$  gjør en rollback.

**Tillates skitten skriv, vil ikke gjenopprettelses-systemer fungere.**

P0:  $w_1(x)..w_2(x)..(c_1$  eller  $a_1)$

# Eksempler på skitten skriv-anomalier

## Integritetsregel: $x=y$

Transaksjoner:

$T_1: x:=1; y:=1$

$T_2: x:=2; y:=2$

Sett fra transaksjonsmanageren:

$T_1: w_1(x)w_1(y)$

$T_2: w_2(x)w_2(y)$

## Ikkeserialiserbar plan:

$[x=y=0] w_1(x) w_2(x) w_2(y) c_2 w_1(y) c_1 [x=2, y=1, x \neq y]$

x og y skrives i forskjellig rekkefølge.

## Ikkegjenopprettbar plan:

$w_1(x) w_2(x) w_2(y) w_1(y) c_2 a_1$

$T_2$ s endringer forsvinner dersom  $T_1$  gjør en rollback, så man må kaskadetilbakerulle  $T_2$  selv om den alt er committet(!), dette strider mot D-en i ACID.

# Skitten les

$T_1$  oppdaterer  $x$ . Deretter leser  $T_2$   $x$  før  $T_1$  committer eller avbryter. Dersom  $T_1$  avbryter, vil  $T_2$  ha lest en verdi som ikke "finnes". I tillegg er det mulig for  $T_2$  å lese inkonsistente data når den leser før  $T_1$  har committet sine endringer.

P1:  $w_1(x)..r_2(x)..(c_1$  eller  $a_1)$

# Eksempel på skitten les-anomali

**Integritetsregel:  $x+y=100$**

Transaksjoner:

$T_1: x:=x-40; y:=y+40$

$r_1(x)w_1(x)r_1(y)w_1(y)$

$T_2: \text{beregner } x+y \text{ (sjekk integritetsregelen)}$

$r_2(x)r_2(y)$

**Eksekveringsplan:**

$[x=y=50, x+y=100]$

$r_1(x)w_1(x)r_2(x)r_2(y)c_2r_1(y)w_1(y)c_1$

$[x=10, y=90, x+y=100]$

$T_2$  har lest en ikke-committet (skitten)  $x$  og konkluderer feilaktig at integritetsregelen er brutt ( $T_2$  beregner  $x+y=60$ )

# Ikke-repeterbar les

$T_1$  leser  $x$ . Deretter oppdaterer eller sletter  $T_2$   $x$ . Dersom  $T_1$  forsøker å lese  $x$  en gang til, vil  $x$  ha en annen verdi eller ikke lenger finnes. Fenomenet opptrer når en transaksjon ( $T_2$ ) kan oppdatere data som er blitt lest av en ennå aktiv transaksjon ( $T_1$ ). Ikke-repeterbare les-anomalier kan oppstå uten at  $T_1$  faktisk leser  $x$  to ganger<sup>1</sup> og selv om  $T_1$  bare leser committede verdier for  $x$ .

P2:  $r_1(x)..w_2(x)..(c_1$  eller  $a_1)$

<sup>1</sup>Se eksempelet under A6, som er et spesialtilfelle av P2



# Eksempel på ikke-repeterbar les-anomali

$T_1: r_1(x) r_1(x) c_1$

$T_2: r_2(x) r_2(x) w_2(x) c_2$

Transaksjonene er begge resultatet av følgende prosedyre:

Finn ledig kinosete. Få bekreftelse fra kunde. Sjekk at setet fortsatt er ledig: Hvis ja, reserver, hvis nei, si at kunden må prøve på nytt.

$T_1$  og  $T_2$  prøver å reservere samme sete  $x$ . Bare  $T_2$  lykkes.

**Eksekveringsplan:**

$[x \text{ er ledig}] r_1(x) r_2(x) r_2(x) w_2(x) c_2 [x \text{ er opptatt}] r_1(x) c_1$

$T_1$  leser  $x$  to ganger og får forskjellig verdi.

*Flere eksempler finnes under skjevlesing (A5A), skjevskrivning (A5B) og lesetransaksjonsanomali (A6), som alle er spesialtilfeller av ikke-repeterbar les-anomalier. Eksempelet over dekkes imidlertid ikke av noen av de tre spesialtilfellene.*

# Tapt oppdatering

$T_1$  leser  $x$ . Deretter oppdaterer  $T_2$   $x$  (muligens basert på en tidligere leseoperasjon). Når så  $T_1$  oppdaterer  $x$  og committer, "forsvinner" oppdateringen gjort av  $T_2$ . Dette er uavhengig av om  $T_2$  committer eller ikke.

P4:  $r_1(x) \dots w_2(x) \dots w_1(x) \dots c_1$

# Eksempel på tapt oppdateringsanomali

$T_1: x := x + 10$        $r_1(x) w_1(x)$   
 $T_2: x := x + 20$        $r_2(x) w_2(x)$

## Eksekveringsplan:

$[x=100]$

$r_1(x) r_2(x) w_2(x) c_2 w_1(x) c_1$

$[x=110]$

$T_2$ s oppdatering går tapt. Etter at både  $T_1$  og  $T_2$  var ferdige, skulle vi hatt  $x=130$ .

# Skjevlesing

Dersom  $T_2$  oppdaterer  $x$  og  $y$  før  $T_1$  har avsluttet sine leseoperasjoner, kan  $T_1$  produsere inkonsistent output.

Skjevlesing er en versjon av ikkereketerbar les (P2) ettersom begge lar  $T_2$  oppdatere data som er blitt lest av en ennå aktiv transaksjon.

A5A:  $r_1(x) \dots w_2(x) \dots w_2(y) \dots c_2 \dots r_1(y) \dots c_1$

Planer som følger A5A, er aldri konfliktserialiserbare.

# Eksempel på skjevlesingsanomali

## Integritetsregel: $x=y$

$T_1$ : les  $x$  og  $y$  (sjekk integritetsregelen)

$r_1(x)r_1(y)$

$T_2$ :  $x:=x+10$ ;  $y:=y+10$

$r_2(x)w_2(x)r_2(y)w_2(y)$

## Eksekveringsplan:

$[x=y=10]$

$r_1(x) r_2(x) w_2(x) r_2(y) w_2(y) c_2 r_1(y) c_1$

$[x=y=20]$

$T_1$  konkluderer feilaktig at integritetsregelen er brutt  
( $T_1$  leser  $x=10$  og  $y=20$ ).

# Skjevskrivning

$T_1$  leser  $x$  og  $y$  som tilfredsstillter en integritetsregel  $C$ .  $T_2$  leser også  $x$  og  $y$ , skriver  $x$  og committer. Dersom  $T_1$  nå skriver  $y$ , kan integritetsregelen  $C$  være brutt.

Skjevskrivning er, som skjevlesing, en versjon av ikkereperterbar les (P2).

A5B:  $r_1(x)..r_2(y)..w_1(y)..w_2(x)..(c_1 \text{ og } c_2)$

Planer som følger A5B, er aldri konfliktserialiserbare.

# Eksempel på skjevskrivingsanomali

**Integritetsregel:  $x \geq y$**

$T_1$ : if  $x > y$  then  $y := y + 1$

$T_2$ : if  $x > y$  then  $x := x - 1$

$r_1(x) \ r_1(y) \ w_1(y)$

$r_2(x) \ r_2(y) \ w_2(x)$

**Eksekveringsplan:**

$[x=10, y=9, x \geq y]$

$r_1(x) \ r_1(y) \ r_2(x) \ r_2(y) \ w_1(y) \ w_2(x) \ c_2 \ c_1$

$[x=9, y=10, x < y]$

Integritetsregelen er brutt.

# Lesetransaksjonsanomali

$T_3$  er en ren lesetransaksjon.  $T_3$  leser  $x$  og  $y$  som oppfyller en integritetsregel  $C$ . To samtidige transaksjoner  $T_1$  og  $T_2$  modifierer henholdsvis  $x$  og  $y$ . Selv om  $C$  bevares, kan  $T_3$  produsere inkonsistent output.

Lesetransaksjonsanomali er også en versjon av ikkereperterbar les (P2).

A6:  $r_2(x) \dots r_2(y) \dots w_1(y) \dots c_1 \dots r_3(x) \dots r_3(y) \dots c_3 \dots w_2(x) \dots c_2$

Planer som følger A6, er aldri konfliktserialiserbare.



# Eksempel på lesetransaksjonsanomali

**Integritetsregel:  $x \geq y$**

$T_1$ :  $y := y - 1$

$T_2$ : if  $x > y$  then (if  $x > y + 1$  then  $x := x - 2$  else  $x := x - 1$ )

$T_3$ : les  $x$  og  $y$

$r_1(y)$   $w_1(y)$

$r_2(x)$   $r_2(y)$   $w_2(x)$

$r_3(x)$   $r_3(y)$

**Eksekveringsplan:**

$[x=10, y=9, x \geq y]$

$r_2(x)$   $r_2(y)$   $r_1(y)$   $w_1(y)$   $c_1$   $r_3(x)$   $r_3(y)$   $c_3$   $w_2(x)$   $c_2$

$[x=9, y=8, x \geq y]$

$T_3$  observerer at  $(x, y) = (10, 8)$ , men dette kan ikke forklares ved  $T_3; T_2; T_1$ ,  $T_2; T_3; T_1$  eller  $T_2; T_1; T_3$ , for da hadde  $T_3$  konkludert med henholdsvis  $(x, y) = (10, 9)$ ,  $(x, y) = (9, 9)$  eller  $(x, y) = (9, 8)$ . Og heller ikke ved  $T_3; T_1; T_2$ ,  $T_1; T_3; T_2$  eller  $T_1; T_2; T_3$ , for da hadde slutttilstanden vært  $x = y = 8$ .

# Fantomfenomenen

Under betegner  $Q$  et *predikat*.  $Q$  identifiserer en mengde av tupler som skal beskyttes mot enhver skriveoperasjon (write, delete, update) etter at  $T_1$  har lest tuplene i  $Q$  og inntil  $T_1$  avslutter. Predikater spesifiseres vanligvis som et uttrykk i where-betingelsen i en SQL **select**, **update** eller **delete**-setning. Fantomfenomener kan oppstå hvis en samtidig transaksjon  $T_2$  øker resultatmengden til  $Q$ .

P3:  $r_1(Q) \dots w_2(y \text{ i } Q) \dots (c_1 \text{ eller } a_1)$

# Fantomleseanomali

$T_1$  finner tuplene som oppfyller  $Q$ . Deretter legger  $T_2$  inn et nytt tuppel som også oppfyller  $Q$ . Når  $T_1$  beregner resultatmengden til  $Q$  en gang til (etter at  $T_2$  har committet), er resultatmengden endret. Fantomleseanomalier kan oppstå også uten at  $T_1$  beregner resultatmengden til  $Q$  to ganger (se eksempelet).

A3A:  $r_1(Q)..w_2(y \ i \ Q)..c_2..r_1(Q)..c_1$

Planer som følger A3A, er aldri konfliktserialiserbare.

# Eksempel på fantomleseanomali

**Integritetsregel:  $x = \#(\text{Ansatt})$**

$T_1$ : sammenlikn resultatet av "**select count(\*) from Ansatt**" med  $x$

$T_2$ : **insert into Ansatt values a;  $x:=x+1$**

Sett fra transaksjonsmanageren: La  $Q = \text{select * from Ansatt}$

$T_1$ :  $r_1(Q) r_1(x)$

$T_2$ :  $w_2(a \text{ i } Q) r_2(x) w_2(x)$

**Eksekveringsplan:**

[#Q=x=100]

$r_1(Q) w_2(a \text{ i } Q) r_2(x) w_2(x) c_2 r_1(x) c_1$

[#Q=x=101]

$T_1$  beregner  $\#(\text{Ansatt})=100$ , men leser  $x=101$  og konkluderer feilaktig med at integritetsregelen er brutt. Merk at  $T_1$  leser Ansatt bare en gang, men likevel kan observere fantomfenomenet (gjennom å lese  $x$ ).

# Fantomskjevskrivning

$T_1$  og  $T_2$  beregner resultatmengden til  $Q$  som tilfredsstillende en integritetsregel  $C$ . Dersom  $T_1$  og  $T_2$  nå hver for seg øker resultatmengden til  $Q$ , kan integritetsregelen  $C$  være brutt.

A3B:  $r_1(Q)..r_2(Q)..w_1(y \ i \ Q)..w_2(z \ i \ Q)..(c_1 \ og \ c_2)$

Planer som følger A3B, er aldri konfliktserialiserbare

# Eksempel på fantomskjevskrivning

**Integritetsregel:** Ansatte kan pålegges maks 8 timer arbeid pr. dag.

Både  $T_1$  og  $T_2$  gjør følgende: Sjekk hvor mange timer Pål er pålagt en gitt dato  $d$ . Legg inn en arbeidsoppgave på 5 timer på Pål denne datoen dersom summen ikke overstiger 8.

Sett fra transaksjonsmanageren:

La  $Q = \text{select } * \text{ from Oppdrag where pid = Pål and dag = } d$

$T_1: r_1(Q) \ w_1((Pål, 5, d) \ i \ Q)$

$T_2: r_2(Q) \ w_2((Pål, 5, d) \ i \ Q)$

**Eksekveringsplan:**

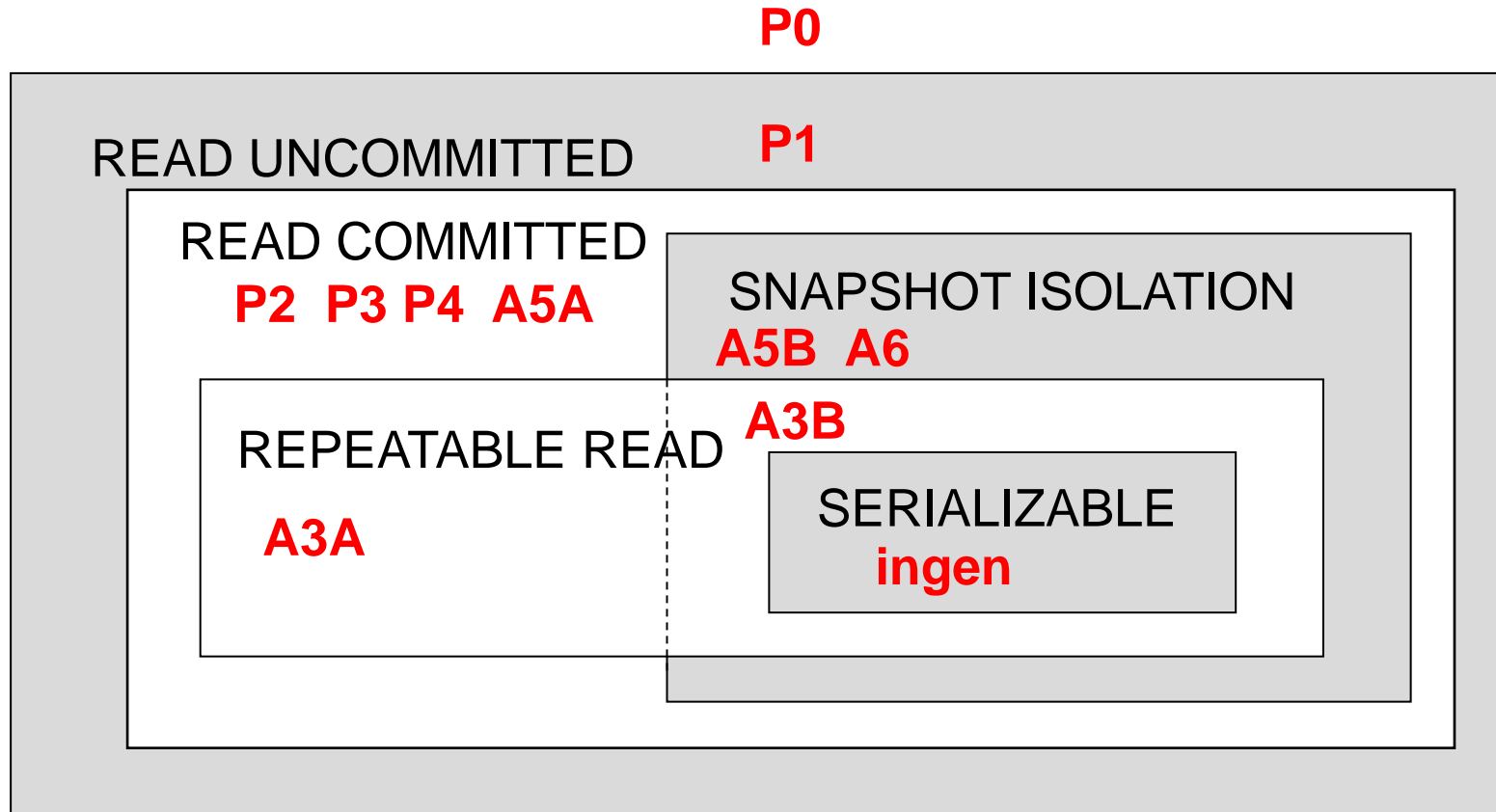
[sum(select timer from Q) = 0]

$r_1(Q) \ r_2(Q) \ w_1((Pål, 5, d) \ i \ Q) \ w_2((Pål, 5, d) \ i \ Q) \ c_1 \ c_2$

[sum(select timer from Q) = 10]

Etter at  $T_1$  og  $T_2$  har fullført, er integritetsregelen brutt.

# Isolasjonsnivåer



Isolasjonsnivåer og hvilke fenomener og anomalier som kan forekomme i hvert av dem