SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Bartul Brajković, 0036507098

1. lipanj 2021.

LABORATORIJ PROFILA 2

Odjeljak Sustavi baza podataka

1. Vježba

2. ZADATAK

Za lakše praćenje sljedećih upita važno je znati id obaju sjednica:

```
Se

lect @@spid -- id sjednice
```

sessionA = 77

sessionB = 84

2.1. Zadatak

T1 (sessionA)	T2 (sessionB)
SET LOCK_TIMEOUT -1;	SET LOCK_TIMEOUT -1;
BEGIN TRANSACTION;	BEGIN TRANSACTION;
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;	SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
SELECT mjera FROM test WHERE sifra = 5;	SELECT mjera FROM test WHERE sifra = 4;
UPDATE test SET mjera = 0 WHERE sifra = 8;	
UPDATE test SET mjera = 0 WHERE sifra = 9;	
UPDATE test SET mjera = 0 WHERE sifra = 10;	
UPDATE test SET mjera = 0 WHERE sifra = 11;	
	UPDATE test SET mjera = 100 WHERE sifra = 9; --čeka na A
	ČEKA T1
UPDATE test SET mjera = 0 WHERE sifra = 4; --čeka na B - potpuni zastoj	

POTPUNI ZASTOJ!!!	
COMMIT TRANSACTION;	Msg 1205, Level 13, State 51, Line 14 Transaction (Process ID 84) was deadlocked on lock resources with another process and has been chosen as the deadlock victim. Rerun the transaction.

Iako je transakcija T1 izazvala potpuni zastoj, SQL Server, kod razrješenja potpunog zastoja, za žrtvu odabire transakciju čije će poništavanje uzrokovati najmanji trošak (T2), a ne transakciju čiji je zahtjev neposredno uzrokovao zatvaranje petlje u WFG grafu (T1).

Budući da je transakcija T1 obavila veću količinu ukupnog posla, iako je ona izazvala potpuni zasoj (petlja u WFG grafu) – poništava se transakcija T2 jer SQL Server odabire transakciju čije će poništavanje uzrokovati obavljanje manjeg broja **undo** operacija (dakle onu koja je obavila manji broj operacija pisanja).

Rezultat izvršavanja upita za provjeru koja transakcija (sjednica) eventualno čeka zbog ključeva koje je postavila neka druga transakcija (sjednica):

```
SELECT session_id AS sid_blocked
 , blocking_session_id AS sid_blocking
FROM sys.dm_os_waiting_tasks
WHERE blocking_session_id IN (SELECT session_id
                              FROM sys.dm_exec_sessions
                              WHERE CAST(context_info AS INT) = 7);
```

Rezultat:

	sid_blocked	sid_blocking
1	84	77
2	77	84

Ako bismo pogledali trenutno stanje naše **test** relacije vidjet ćemo da je uistinu transakcija T1 obavila upit: `UPDATE test SET mjera = 0 WHERE sifra = 4;`

Također možemo vidjeti da se posljednji upit transakcije T2 nije uzvršio: `UPDATE test SET mjera = 100 WHERE sifra = 9;`

	sifra	mjera
1	1	10.0
2	2	20.0
3	3	30.0
4	4	0.0
5	5	50.0
6	6	60.0
7	7	70.0
8	8	0.0
9	9	0.0
10	10	0.0
11	11	0.0
12	12	120.0

2.2. Zadatak

Cilj drugog zadatka bio je pokazati kako se podešavanjem prioriteta korisničke sjednice (*priority of session in deadlock situation*) može utjecati na to da neka transakcija, bez obzira na njezin relativno mali trošak poništavanja, preživi razrješenje potpunog zastoja na štetu neke druge transakcije čiji je trošak poništavanja relativno velik.

To sam učinio tako da sam neposredno nakon pokretanja transakcije postavio prioritete korisničkim sjednicama. Jedina razlika je bila što sam transakciji T1 prioritet postavio na LOW, a transakciji T2 prioritet postavio na HIGH kao što je prikazano u nastavku:

T1:

```
SET LOCK_TIMEOUT -1;
BEGIN TRANSACTION;
SET DEADLOCK_PRIORITY LOW; -- 2.2. ZADATAK
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
```

T2:

```
SET LOCK_TIMEOUT -1;
BEGIN TRANSACTION;
SET DEADLOCK_PRIORITY HIGH; -- 2.2. ZADATAK
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
```

Ostale upite nisam mijenjao (isti su onima iz 2.1. zadatka).

Sada nakon izvršavanja upita transakcija T1 i T2 (istim redosljedom kao i u 2.1. dijelu zadatka) dobivam različiti rezultat.

Budući da je na transakcija T2 većeg prioriteta od transakcije T1, sada će T1 biti poništena iako je napravila veću količinu ukupnog posla.

Poruka od T1:

```
Msg 1205, Level 13, State 45, Line 21
Transaction (Process ID 77) was deadlocked on lock resources with another process and has been chosen as the deadlock victim. Rerun the transaction.
```

Trenutno stanje relacije **test**:

	sifra	mjera
1	1	10.0
2	2	20.0
3	3	30.0
4	4	40.0
5	5	50.0
6	6	60.0
7	7	70.0
8	8	80.0
9	9	100.0
10	10	100.0
11	11	110.0
12	12	120.0

Iz trenutnog stanja relacije možemo vidjeti kako je uistinu transakcija T1 poništena, dok je ovaj put transakcija T2 izvršena.

3. ZADATAK

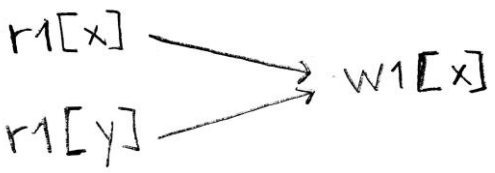
Transakcije:

a)
```
sumirajXY(x, y)
    read(x, p1);
    read(y, p2);
    p3 ← p1 + p2;
    write(x, p3);
```

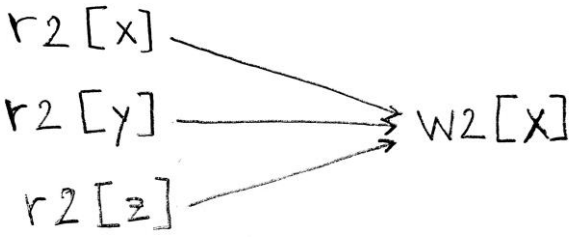
b)
```
sumirajXYZ(x, y, z)
    read(x, p1);
    read(y, p2);
    read(z, p3);
    p4 ← p1 + p2 + p3;
    write(x, p4);
```

3.1. Zadatak

T1:



T2:

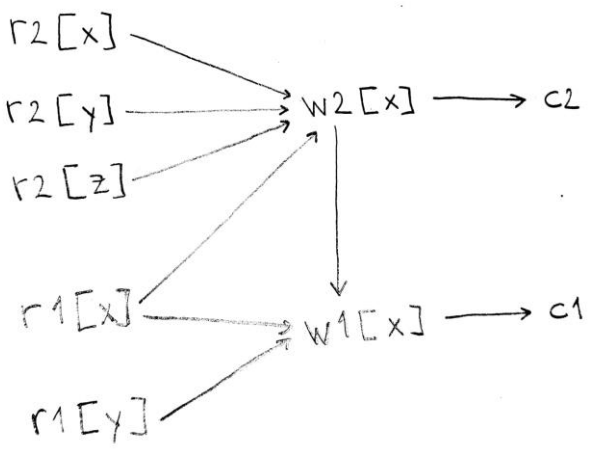


3.2. Zadatak

Povijest koja dovodi do potpunog zastoja ako SUBP ne koristi U-ključeve, a ne dovodi do potpunog zastoja ako SUBP koristi U-ključeve:

H: r2[x], r2[y], r2[z], r1[x], r1[y], w2[x], w1[x], c2, c1

Graf:

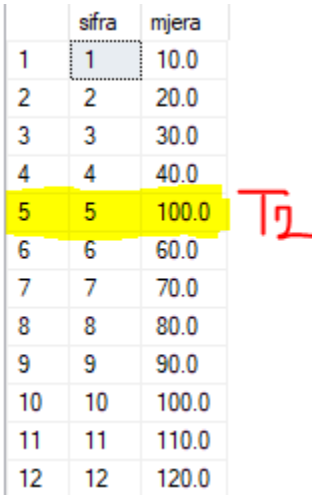


3.3. Zadatak

T1 (sessionA)	T2 (sessionB)
`SET LOCK_TIMEOUT -1;`	`SET LOCK_TIMEOUT -1;`
`BEGIN TRANSACTION;`	`BEGIN TRANSACTION;`
`SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;`	`SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;`
	`SELECT mjera FROM test WHERE sifra = 5;`
	`SELECT mjera FROM test WHERE sifra = 12;`
	`SELECT mjera FROM test WHERE sifra = 17;`

`SELECT mjera FROM test WHERE sifra = 5;`	
`SELECT mjera FROM test WHERE sifra = 12;`	
	`UPDATE test SET mjera = 100 WHERE sifra = 5;`
	T2 čeka
`UPDATE test SET mjera = 0 WHERE sifra = 5;`	
POTPUNI ZASTOJ!!!	
Msg 1205, Level 13, State 51, Line 37 Transaction (Process ID 77) was deadlocked on lock resources with another process and has been chosen as the deadlock victim. Rerun the transaction.	`COMMIT TRANSACTION;`

Ako pogledamo trenutno stanje relacije **test** možemo vidjeti da je zaista transakcija T2 uspješno izvršena dok je transakcija T1 poništena:



	sifra	mjera
1	1	10.0
2	2	20.0
3	3	30.0
4	4	40.0
5	5	100.0
6	6	60.0
7	7	70.0
8	8	80.0
9	9	90.0
10	10	100.0
11	11	110.0
12	12	120.0

T2

4. ZADATAK

4.1. Zadatak

Transakcije:

```
a) sumirajXY(x, y)          -- u x zapiši sumu vrijednosti x i y, oznaka transakcije T1
      read(x, p1);
      read(y, p2);
      p3 ← p1 + p2;
      write(x, p3);

b) pisiXY(x, y)             -- u x i y zapiši konstante 101.0 i 102.0, oznaka transakcije T2
      write(x, 101.0);
      write(y, 102.0);
```

Povijest koja dovodi do potpunog zastoja:

H: w2[x], r1[y], r1[x], w2[y], w1[x]

Ako Ti zaključa element x U-ključem:

- Ti može čitati x
- Ti može U-ključ promovirati u X-ključ (pod uvjetom da u trenutku promocije x nije zaključan S-ključevima drugih transakcija)

4.2. Zadatak

T1 (sessionA)	T2 (sessionB)
`SET LOCK_TIMEOUT -1;`	`SET LOCK_TIMEOUT -1;`
`BEGIN TRANSACTION;`	`BEGIN TRANSACTION;`
`SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;`	`SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;`
	`UPDATE test SET mjera = 100 WHERE sifra = 3;`

`SELECT mjera FROM test WHERE sifra = 9;`	
`SELECT mjera FROM test WHERE sifra = 3;`	
	`UPDATE test SET mjera = 100 WHERE sifra = 9;`
POTPUNI ZASTOJ!!!	
`Msg 1205, Level 13, State 45, Line 34 Transaction (Process ID 77) was deadlocked on lock resources with another process and has been chosen as the deadlock victim. Rerun the transaction.`	`COMMIT TRANSACTION;`

Ovakav poredak transakcija uzrokovao je potpuni zastoj pri kojemu je transakcije T1 poništena dok je transakcija T2 uspješno izvršena.

Trenutno stanje relacije **test** je:

	sifra	mjera	
1	1	10.0	
2	2	20.0	
3	3	100.0	T2
4	4	40.0	
5	5	50.0	
6	6	60.0	
7	7	70.0	
8	8	80.0	
9	9	100.0	T2
10	10	100.0	
11	11	110.0	
12	12	120.0	

5. ZADATAK

T_1 (sessionA)	T_2 (sessionB)
`SET LOCK_TIMEOUT -1;`	`SET LOCK_TIMEOUT -1;`
`BEGIN TRANSACTION;`	`BEGIN TRANSACTION;`
`SET TRANSACTION ISOLATION LEVEL` ***xlevel***`;`	`SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;`
1. `SELECT mjera FROM test WHERE sifra = 5;`	
	2. `SELECT mjera FROM test WHERE sifra = 8;`
	3. `UPDATE test SET mjera = 52.0 WHERE sifra = 8;`
	4. `SELECT mjera FROM test WHERE sifra = 5;`
	5. `UPDATE test SET mjera = 82.0 WHERE sifra = 5;`

5.1. xlevel - READ UNCOMMITTED

1. T1 postavlja ključ - ne
2. T2 postavlja IS na relaciju test – da; S na zapis indeksa sifra = 8 – da; S na n-torku sifra = 8 – da;
3. T2 postavlja IX na relaciju test – da; X na zapis indeksa sifra = 8 -da; X na n-torku sifra = 8 – da;
4. T2 postavlja IX na relaciju test – da; X na zapis indeksa sifra = 5 – da; X na n-torku sifra = 5 – da;

5.2 . xlevel – READ COMMITED

1. T1 postavlja IS ključ na relaciju test – da; S na zapis indeksa sifra = 5 – da; S na n-torku sifra =5 – da;
2. T1 otpusti ključeve
3. T2 postavlja IS na relaciju test – da; S na zapis indeksa sifra = 8 – da; S na n-torku sifra =8 – da;
4. T2 postavlja IX na relaciju test – da; X na zapis indeksa sifra = 8 – da, X na n-torku sifra=8 – da;
5. T2 postavlja IX na relaciju test – da; X na zapis indeksa sifra = 5 – da; X na n-torku sifra=5 – da;

5.3. xlevel – SERIALIZABLE

1. T1 postavlja IS na relaciju test – da; S na zapis indeksa sifra = 5 – da; S na n-torku sifra =5 – da;
2. T2 postavlja IS na relaciju test – da; S na zapis indeksa sifra = 8 – da; S na n-torku sifra =8 – da;
3. T2 postavlja IX na relaciju test – da; X na zapis indeksa sifra = 8 – da; X na n-torku sifra=8 – da;
4. T2 postavlja IX na relaciju test – da; X na zapis indeksa sifra = 5 – NE; X na n-torku sifra=5 – NE;

T2 ne postavlja ključ X na zapis indeksa zato što transakcija T1 još uvijek drži S ključ na n-torci sa šifrom 5.

5.4. Xlevel – READ UNCOMITTED uništen je indeks nad atributom sifra u relaciji test

1. T1 postavlja ključ – NE
2. T2 postavlja S na relaciju test – DA
3. T2 postavlja X na relaciju test – DA
4. T2 postavlja X na relaciju test – DA

5.5. Xlevel – READ COMMITED uništen je indeks nad atributom sifra u relaciji test

1. T1 postavlja S na relaciju test – DA
2. T1 otpušta ključ
3. T2 postavlja S na relaciju test – DA
4. T2 postavlja X na relaciju test – DA
5. T2 postavlja X na relaciju test – DA

5.6. Xlevel – SERIALIZABLE

1. T1 postavlja S na relaciju test – DA
2. T2 postavlja S na relaciju test – DA
3. T2 postavlja X na relaciju test – NE
4. T2 postavlja X na relaciju test – NE

T2 neće postaviti svoje X ključeve zato što transakcija T1 još uvijek drži S ključ nad relacijom test.