SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Bartul Brajković, 0036507098

26. svibanj 2021.

LABORATORIJ PROFILA 2

Odjeljak Sustavi baza podataka

8. Vježba

1. ZADATAK

1.1. Zadatak

```
select @@spid -- id sjednice
```

sessionA = 53

sessionB = 61

sessionC = 62

sessionD = 63

sessionAdmin = 65

1.2. Zadatak

```
SELECT session_id, login_time, DB_NAME(database_id) AS DB
FROM sys.dm_exec_sessions
```

53	2021-05-26 14:51:11.567	labprof8
54	2021-05-26 15:05:45.830	master
55	2021-05-26 15:05:45.830	master
56	2021-05-26 14:59:45.190	master
57	2021-05-26 15:05:45.830	master
58	2021-05-26 15:05:45.830	master
59	2021-05-26 15:05:09.803	master
60	2021-05-26 15:05:45.830	master
61	2021-05-26 14:51:11.910	labprof8
62	2021-05-26 14:51:12.353	labprof8
63	2021-05-26 14:51:12.817	labprof8
64	2021-05-26 15:05:45.830	master
65	2021-05-26 15:01:14.290	labprof8

1.3. Zadatak

```
SELECT session_id, login_time, DB_NAME(database_id) AS DB
FROM sys.dm_exec_sessions
WHERE CONTEXT_INFO = 7;
```

	session_id	login_time	DB
1	53	2021-05-26 14:51:11.567	labprof8
2	61	2021-05-26 14:51:11.910	labprof8
3	62	2021-05-26 14:51:12.353	labprof8
4	63	2021-05-26 14:51:12.817	labprof8

2. ZADATAK

2.1. Zadatak

SET LOCK_TIMEOUT -1

- Postavlja LOCK_TIMEOUT da čeka zauvijek
- LOCK_TIMEOUT predstavlja zabranu

Kada jedan korisnik zaključa neke podatke i zadrži ih dok drugi korisnik pokušava pristupiti njima. Ako prvi korisnik ne otključa podatke, drugi će nakon nekog vremena isteći. Baza podataka odgovorit će drugom korisniku porukom pogreške u kojoj se kaže da je čekanje zaključavanja predugo.

2.2. Zadatak

T_1 (sessionA)	T_2 (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 = 8;`
	`UPDATE test SET mjera = 82.0 WHERE sifra = 8;`
`UPDATE test SET mjera = 81.0 WHERE sifra = 8;`	
`ROLLBACK TRANSACTION;`	`ROLLBACK TRANSACTION;`

sessionA: `SELECT mjera FROM test WHERE sifra = 5;`

	sid	lock_type	sifra
1	53	S	5

sessionB: `SELECT mjera FROM test WHERE sifra = 8;`

	sid	lock_type	sifra
1	53	S	5
2	61	S	8

sessionB: `UPDATE test SET mjera = 82.0 WHERE sifra = 8;`

	sid	lock_type	sifra
1	53	S	5
2	61	X	8

sessionA: `UPDATE test SET mjera = 81.0 WHERE sifra = 8;`

	sid	lock_type	sifra
1	61	X	8

sessionA: `Transaction (Process ID 53) was deadlocked on lock resources with another process and has been chosen as the deadlock victim. Rerun the transaction.`

Za provjeru koji su ključevi postavljeni na koje n-torke relacije koristi se:

sessionAdmin:

```
SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;

SELECT request_session_id AS sid
 , CASE WHEN request_mode = 'U' THEN 'S' ELSE request_mode END AS lock_type
 , test.sifra
 FROM sys.dm_tran_locks
 LEFT OUTER JOIN test
 ON sys.fn_PhysLocFormatter(%%physloc%%) = '(' + TRIM(resource_description) + ')'
 WHERE resource_type = 'RID'
 AND request_status = 'GRANT'
 AND request_session_id IN (SELECT session_id FROM sys.dm_exec_sessions where
CAST(context_info AS INT) = 7);
```

2.3. Zadatak

Transakcija B s naredbom UPDATE je postavila ključ X na n-torci sa šifrom 8.

Transakcija A s naredbom UPDATE nije postavila ključ jer čeka da se oslobodi ključ sa šifrom 8.

Transakcija A čeka.

Nije nastao potpuni zastoj zato što kad transakcija B završi ona će otpustiti ključ, tada transakcija A može nastaviti.

2.4. Zadatak

Transakcija T2 je uspjela izvršiti UPDATE nad n-torkom koja je zaključana S ključem zato što ako transakcija postavi S ključ, ista ta transakcija smije ga pretvorit u X ključ.

3. ZADATAK

```
IF OBJECT_ID('proc3') IS NOT NULL DROP PROCEDURE proc3


 CREATE PROCEDURE proc3 @sifra INTEGER
 AS
       BEGIN TRANSACTION
       SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
       BEGIN TRY

       --test
              SET LOCK_TIMEOUT -1;
              UPDATE test SET mjera = mjera * 1.1 WHERE test.sifra = @sifra;

       --test2
              SET LOCK_TIMEOUT 5000;
              UPDATE test2 SET mjera = mjera * 1.1 WHERE test2.sifra = @sifra;

       --test3
              SET LOCK_TIMEOUT 0;
              UPDATE test3 SET mjera = mjera * 1.1 WHERE test3.sifra = @sifra;

       END TRY

       BEGIN CATCH
              ROLLBACK TRANSACTION

              IF (ERROR_MESSAGE() LIKE '%Lock%')
                     THROW 50501, 'Privremeno zaključano, pokušajte kasnije', 1;
              ELSE
                     THROW
       END CATCH

COMMIT TRANSACTION
GO


EXEC dbo.proc3 @sifra = 2;

SELECT * FROM test;
SELECT * FROM test2;

SELECT * FROM test3;
```

	sifra	mjera
1	1	10.0
2	2	22.0
3	3	30.0
4	4	40.0
5	5	50.0

4. ZADATAK

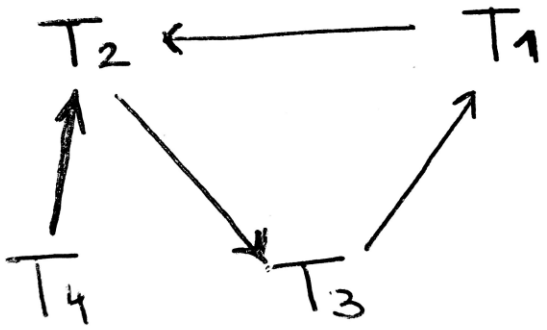
4.1. Zadatak

T1 (sessionA)	T2 (sessionB)	T3 (sessionC)	T4 (sessionD)
SET LOCK_TIMEOUT -1;	SET LOCK_TIMEOUT -1;	SET LOCK_TIMEOUT -1;	SET LOCK_TIMEOUT -1;
BEGIN TRANSACTION;	BEGIN TRANSACTION;	BEGIN TRANSACTION;	BEGIN TRANSACTION;
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;	SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;	SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;	SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
		UPDATE test2 SET mjera = 2.2 WHERE sifra = 8	
	UPDATE test2 SET mjera = 8.8 WHERE sifra = 8		
			SELECT mjera FROM test2 WHERE sifra = 8
SELECT mjera FROM test2 WHERE sifra = 5			
	UPDATE test2 SET mjera = 5.5 WHERE sifra = 5		
		UPDATE test2 SET mjera = 9.9 WHERE sifra = 9	
SELECT mjera FROM test2 WHERE sifra = 9			
ROLLBACK TRANSACTION;	ROLLBACK TRANSACTION;	ROLLBACK TRANSACTION;	ROLLBACK TRANSACTION;

4.2. Zadatak

4.2.

NFG:



4.3. Zadatak

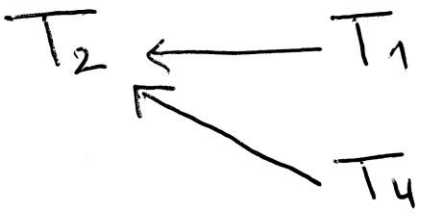
SUBP rješava problem potpunog zastoja:

- izbjegavanjem potpunih zastoja; npr. protokol se dopunjava pravilima koja onemogućuju pojavu potpunih zastoja (npr. zabrana čekanja)
- detekcijom te poništavanjem jedne ili više transakcija

4.4 Zadatak

4.4.

WFG:

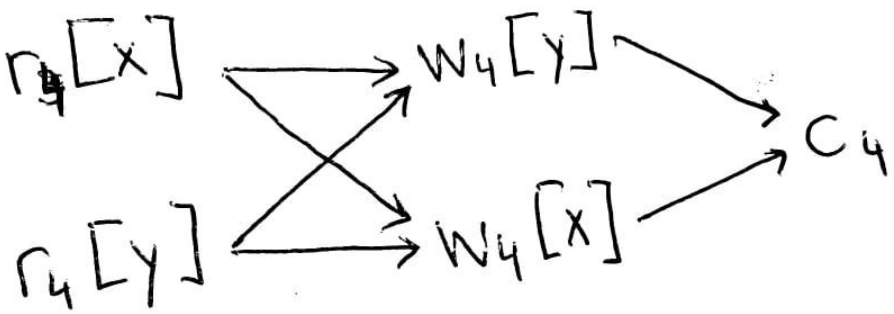


a) $w_1[x] \longrightarrow c_1$

b) $r_2[x] \longrightarrow c_2$

c) $r_3[x] \longrightarrow w_3[x] \longrightarrow c_3$

d)



e)