

## 9. VJEŽBA

### 1. zadatak

Procjena potrebnog vremena: 5 minuta

U rješenjima je potrebno dostaviti:

- ništa

Preuzeti datoteku i izvršiti skripta `labprof9.sql`.

Otvoriti dvije SQL Editor kartice (*tab*), a u svakoj od kartica korisničku sjednicu (*session*) za korisničko ime "sa" i bazu podataka labprof9, te u svakoj od tih korisničkih sjednica obaviti naredbu `SET CONTEXT_INFO 7`;

Otvoriti još jednu (treću) korisničku sjednicu u kojoj će se moći postavljati upiti koji služe za prikaz trenutačno postavljenih ključeva.

### Pomoćni upiti za zadatke u ovim laboratorijskim vježbama

U ovim laboratorijskim vježbama, za provjeru koji ključevi su postavljeni na koje objekte povezane s relacijom `test`, može se koristiti sljedeći upit:

```
SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
SELECT request_session_id AS sid
      , 'KEY' AS object
      , request_mode AS lock_type
      , CAST (test.sifra AS VARCHAR) AS identifikator
FROM sys.dm_tran_locks
LEFT OUTER JOIN test WITH(NOLOCK INDEX(ind1Test))
  ON test.%%lockres%% = resource_description
WHERE resource_type = 'KEY'
      AND request_status = 'GRANT'
      AND request_session_id IN (SELECT session_id FROM sys.dm_exec_sessions where CAST(context_info AS INT) = 7)
UNION
SELECT request_session_id AS sid
      , 'ROW' AS object
      , request_mode AS lock_type
      , CAST (test.sifra AS VARCHAR) AS identifikator
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)
UNION
SELECT request_session_id AS sid
      , 'TABLE' AS object
      , request_mode AS lock_type
      , sys.objects.name
FROM sys.dm_tran_locks
LEFT OUTER JOIN sys.objects
  ON sys.objects.object_id = sys.dm_tran_locks.resource_associated_entity_id
WHERE resource_type = 'OBJECT'
      AND request_status = 'GRANT'
      AND request_session_id IN (SELECT session_id FROM sys.dm_exec_sessions where CAST(context_info AS INT) = 7);
```

Za provjeru koja transakcija (sjednica) eventualno čeka zbog ključeva koje je postavila neka druga transakcija (sjednica), može se koristiti sljedeći upit:

```
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);
```

## 2. zadatak

Procjena potrebnog vremena: 30 minuta

U rješenjima je potrebno dostaviti:

- Rješenja zadataka 2.1. i 2.2.

Tijekom razrješenja potpunog zastoja neki sustavi za transakciju-žrtvu jednostavno odaberu transakciju čiji je zahtjev neposredno uzrokovao zatvaranje petlje u WFG grafu. Međutim, SQL Server (uz uvjet da korisničke sjednice u kojima se izvršavaju transakcije koje su ušle u potpuni zastoj imaju jednaki prioritet, *priority of session in deadlock situation*) za žrtvu odabire transakciju čije će poništavanje uzrokovati najmanji trošak.

- 2.1. Eksperimentima pokazati da će SQL Server, kod razrješenja potpunog zastoja, za žrtvu odabrati transakciju čije će poništavanje uzrokovati najmanji trošak, a ne transakciju čiji je zahtjev neposredno uzrokovao zatvaranje petlje u WFG grafu.
- 2.2. Eksperimentima 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.

Pri tome, u oba zadatka slobodno odabrati nad kojim n-torkama relacije test će se izvršavati koje operacije transakcija.

Tijekom nadziranih provjera/vježbi očekuju se sljedeća znanja i vještine:

- Objasniti vlastita rješenja. Riješiti sličan zadatak, testirati i objasniti rezultate.

## 3. zadatak

Procjena potrebnog vremena: 30 minuta

U rješenjima je potrebno dostaviti:

- Rješenja zadataka 3.1. - 3.3.

U sustavu se izvršava jedna transakcija vrste sumirajXY i jedna transakcija vrste sumirajXYZ, koje su opisane sljedećim pseudokodom:

```
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) **sumirajXYZ(x, y, z)** -- u x zapiši sumu vrijednosti x, y i z, oznaka transakcije T<sub>2</sub>

```

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

```

- 3.1. Nacrtati grafove transakcija T<sub>1</sub> i T<sub>2</sub>. U svaki od grafova transakcija ucrtati najmanji mogući broj lukova: isključivo one lukove koji proizlaze iz semantike transakcije ili su nužni da bi se zadovoljila pravila konstrukcije grafa transakcije.
- 3.2. U obliku niza operacija (npr. r<sub>1</sub>[x], r<sub>1</sub>[y], w<sub>1</sub>[x], r<sub>2</sub>[x], ...) napisati povijest H koja sadrži transakcije T<sub>1</sub> i T<sub>2</sub>. Povijest treba biti takva da dovodi do potpunog zastoja ako SUBP ne koristi U-ključeve, a ne dovodi do potpunog zastoja ako SUBP koristi U-ključeve. Zatim nacrtati graf povijesti koji je konzistentan s H.
- 3.3. Za nizove naredbi u rješenju zadatka 3.2. načiniti tablice koje su slične tablici u 2. zadatku 8. vježbi. Pri tome, kao vrijednosti x, y i z koristiti vrijednosti atributa *mjera* u n-torkama sa šiframa 5, 12 i 17. Izvršavanjem SQL naredbi dokazati ispravnost svojeg rješenja zadatka 3.2.

Tijekom nadziranih provjera/vježbi očekuju se sljedeća znanja i vještine:

- Objasniti vlastita rješenja. Riješiti sličan zadatak, testirati i objasniti rezultate.

## 4. zadatak

Procjena potrebnog vremena: 20 minuta

U rješenjima je potrebno dostaviti:

- Rješenja zadataka 4.1. i 4.2.

U sustavu se izvršava jedna transakcija vrste *sumirajXY* i jedna transakcija vrste *pisiXY*, koje su opisane sljedećim pseudokodom:

a) **sumirajXY(x, y)** -- u x zapiši sumu vrijednosti x i y, oznaka transakcije T<sub>1</sub>

```

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 T<sub>2</sub>

```

write(x, 101.0);
write(y, 102.0);

```

- 4.1. U obliku niza operacija (npr. r<sub>1</sub>[x], r<sub>1</sub>[y], w<sub>1</sub>[x], r<sub>2</sub>[x], ...) napisati povijest H koja sadrži transakcije T<sub>1</sub> i T<sub>2</sub>. Povijest treba biti takva da dovodi do potpunog zastoja bez obzira koristi li SUBP U-ključeve ili ne.
- 4.2. Za niz naredbi u rješenju zadatka 4.1. načiniti tablice koje su slične tablici u 2. zadatku 8. vježbi. Pri tome, kao vrijednosti x i y koristiti vrijednosti atributa *mjera* u n-torkama sa šiframa 3 i 9. Izvršavanjem SQL naredbi dokazati ispravnost svojeg rješenja zadatka 4.1.

Tijekom nadziranih provjera/vježbi očekuju se sljedeća znanja i vještine:

- Objasniti vlastita rješenja. Riješiti sličan zadatak, testirati i objasniti rezultate.

## 5. zadatak

Procjena potrebnog vremena: 40 minuta

U rješenjima je potrebno dostaviti:

- Rješenja zadataka 5.1. - 5.6.

U sustavu se izvršavaju transakcije  $T_1$  i  $T_2$ .

| $T_1$ (sessionA)                                | $T_2$ (sessionB)                                    |
|---|---|
| SET LOCK TIMEOUT -1;                            | SET LOCK TIMEOUT -1;                                |
| BEGIN TRANSACTION;                              | BEGIN TRANSACTION;                                  |
| SET TRANSACTION ISOLATION LEVEL <b>xlevel</b> ; | SET TRANSACTION ISOLATION LEVEL<br>SERIALIZABLE;    |
| 1. SELECT mjera FROM test WHERE sifra = 5;      |   |
|   | 2. SELECT mjera FROM test WHERE sifra = 8;          |
|   | 3. UPDATE test SET mjera = 52.0<br>WHERE sifra = 8; |
|   | 4. SELECT mjera FROM test WHERE sifra = 5;          |
|   | 5. UPDATE test SET mjera = 82.0<br>WHERE sifra = 5; |

SUBP koristi MGL protokol (uz hijerarhiju objekata: relacija  $\rightarrow$  n-torka) i protokol zaključavanja indeksa, ali samo ako su za korištenje protokola zaključavanja indeksa ostvareni potrebni preduvjeti.

- a) Za svaku SELECT i UPDATE naredbu koja se pokušava izvršiti, napisati kojom vrstom ključa se pokušava zaključati (ili otključati) koji element baze podataka, te ako zaključavanje nije uspjelo, navesti razlog zašto.

Potrebno je napisati rješenja za tri slučaja:

- Ako  $T_1$  na mjestu označenom s **xlevel** koristi razinu izolacije READ UNCOMMITTED?
- Ako  $T_1$  na mjestu označenom s **xlevel** koristi razinu izolacije READ COMMITTED?
- Ako  $T_1$  na mjestu označenom s **xlevel** koristi razinu izolacije SERIALIZABLE?

**Rješenja se trebaju temeljiti na teorijskim postavkama izloženim na predavanjima** i trebaju biti napisana u sljedećem obliku:

### 5.1. za READ UNCOMMITTED

- $T_1$  postavlja SIX na relaciju test - da; X na zapis indeksa sif=100 - da; IX na n-torku sif=100
- $T_2$  postavlja X na relaciju test - da; S na zapis indeksa sif=101 - ne, jer  $T_1$  drži IX na n-torci sif=100  
... itd.

### 5.2. za READ COMMITTED

- $T_1$  postavlja X na relaciju test - da; SIX na zapis indeksa sif=100 - da; otpušta X s relacije test
- $T_2$  postavlja SIX na relaciju test - da; IS na zapis indeksa sif=101 - da; S na n-torku sif=101  
... itd.

Svoja rješenja provjeriti izvršavanjem operacija transakcija  $T_1$  i  $T_2$ . U rješenjima **napisati napomenu** na mjestima na kojima uočite da sustav postavlja drugačije ključeve nego bi se očekivalo prema teorijskim postavkama izloženim na predavanjima.

- b) Uništiti indeks *ind1Test* (indeks nad atributom sifra u relaciji test), te ponoviti zadatak pod a), s time da se rješenja, umjesto s 5.1, 5.2, i 5.3, označe s 5.4, 5.5, i 5.6.

Na kraju, vratiti staro stanje, tj. kreirati indeks *ind1Test* (indeks nad atributom sifra u relaciji test):

```
CREATE UNIQUE INDEX ind1Test ON test (sifra) WITH (ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = OFF);
```

Tijekom nadziranih provjera/vježbi očekuju se sljedeća znanja i vještine:

- Objasniti vlastita rješenja. Riješiti sličan zadatak, testirati i objasniti rezultate.