

Zadaci za vježbu

(uz predavanje 17- Kontrola istodobnog pristupa)

Vrste ključeva:

1. Ključ za čitanje (*READ LOCK*) se postavlja čitanjem n-torke, ovisno o razini izolacije. Na zaključavanje za čitanje **utječe** razina izolacije.
2. Ključ za pisanje (*WRITE LOCK*) se postavlja obavljanjem SQL naredbe koja ažurira podatke (*UPDATE*, *INSERT*, *DELETE*) na objekt kojeg mijenja/briše/unosi. Na zaključavanje za pisanje **ne utječe** razina izolacije, odnosno, postavljanje te vrste ključa ne može se spriječiti, neovisno o postavljenoj razini izolacije. Uklanja se tek po završetku transakcije.

Svaka SQL naredba koja se nalazi izvan eksplicitno zadanih granica transakcija, implicitno se smatra zasebnom transakcijom. Obavit će se u potpunosti (za sve n-torke koje zadovoljavaju uvjet) ili se uopće neće obaviti.

Sljedeći niz naredbi (odnose se na studAdmin bazu podataka):

```
UPDATE student SET spol = 'M'
WHERE substring(JMBG from 10 for 3)< 500;
INSERT INTO zupanija VALUES (1, 'Zupanija 1');
DELETE FROM upisanPredmet WHERE ocjena = 1;
```

se može se promatrati kao:

```
BEGIN WORK;
UPDATE student SET spol = 'M'
WHERE substring(JMBG from 10 for 3)< 500;
ako nema pogreške COMMIT WORK, inače ROLLBACK WORK;
```

```
BEGIN WORK;
INSERT INTO zupanija VALUES (1, 'Zupanija 1');
ako nema pogreške COMMIT WORK, inače ROLLBACK WORK;
```

```
BEGIN WORK;
DELETE FROM upisanPredmet WHERE ocjena = 1;
ako nema pogreške COMMIT WORK, inače ROLLBACK WORK;
```

Granulacija zaključavanja

S ciljem sprječavanja postavljanja prevelikog broja ključeva za pisanje i nepotrebnog ograničavanja konkurentnosti potrebno je koristiti odgovarajuću granulaciju zaključavanja. Poželjno je koristiti čim kraće transakcije (koliko je to moguće s obzirom na pravila integriteta podataka)

Koju granulaciju zaključavanja koristiti?

- DATABASE – ako je potrebno zaključati vrlo veliki broj n-torki u vrlo velikom broju relacija
- TABLE – ako je potrebno zaključati vrlo veliki broj n-torki u ograničenom broju relacija. LOCK TABLE naredba će završiti neuspjehom ako je tablica *imeTablice*:
 - već zaključana ekskluzivno (EXCLUSIVE) nekim drugim procesom ili
 - se pokušava zaključati ekskluzivno, a nekim drugim procesom je već zaključana u dijeljenom (SHARE) modu
- PAGE – ako je potrebno zaključati ograničeni broj n-torki za koje se može pretpostaviti da su smještene na fizički bliskim pozicijama
 - ROW – ako je potrebno zaključati ograničen broj n-torkiPrimjer: U relaciju *racun* unese se "glava računa", a u relaciju *stavke* npr. desetak "stavki" računa. Ako se stavke tog računa unose "odjednom", može se pretpostaviti da će biti pohranjene zajedno, na jednoj ili nekoliko memorijskih stranica. Za relaciju *stavke* koristiti PAGE granulaciju, a za relaciju *racun* ROW granulaciju.
- INDEX, KEY - koristi se implicitno od strane SUBP-a. SUBP zaključava pojedine elemente B-stabla (zapis, blok, indeksnu granu) s ciljem sprečavanja pojave sablasnih n-torki.

Zadatak 1.

Koristeći SQL Editor (Server Studio), kao korisnik *bpadmin*, uspostavite dvije korisničke sjednice s bazom podataka *studAdmin*.

- a) Pokušajte u obje sjednice obaviti naredbu:
DATABASE *studAdmin* EXCLUSIVE;
Je li naredba uspješno izvedena? Obrazložite.
- b) Zatvorite jednu od sjednica i u drugoj (trenutno jednoj) sjednici pokušajte obaviti naredbu:
DATABASE *studAdmin* EXCLUSIVE;
Je li naredba uspješno izvedena? Obrazložite.

Zatvorite obje sjednice.

Zadatak 2.

Koristeći SQL Editor (Server Studio), kao korisnik *bpadmin*, uspostavite dvije korisničke sjednice s bazom podataka *studAdmin*.

U prvoj sjednici izvedite sljedeće naredbe:

```
BEGIN WORK;  
LOCK TABLE upisanPredmet IN SHARE MODE;
```

- a) U drugoj sjednici izvedite sljedeće naredbe:
BEGIN WORK;
LOCK TABLE upisanPredmet IN SHARE MODE;
Jesu li naredbe uspješno izvedene? Obrazložite.

U drugoj sjednici obavite naredbu:

```
ROLLBACK WORK;
```

- b) U drugoj sjednici obavite sljedeće naredbe:
BEGIN WORK;
LOCK TABLE upisanPredmet IN EXCLUSIVE MODE;

Jesu li naredbe uspješno izvedene? Obrazložite.

SELECT naredba u interaktivnom alatu (npr. Server Studio)

Za vrijeme obavljanja SELECT naredbe koja se nalazi izvan eksplicitno zadanih granica transakcije, ključevi za čitanje se postavljaju u skladu s razinom izolacije, ali se eventualno postavljeni ključevi otpuštaju odmah nakon čitanja n-torke (jer se n-torke dohvaćaju izvan granica transakcije).

Ako su granice transakcije eksplicitno zadane, ključevi se ne otpuštaju odmah nakon završetka SELECT naredbe, nego na pročitanim n-torkama (ili stranicama) ostaju ključevi ovisno o postavljenoj razini izolacije.

Slijede objašnjenja u vezi ključeva za čitanje koji se postavljaju pri obavljanju iste SELECT naredbe pri različitim razinama izolacije u IBM Informix SUBP-u (READ UNCOMMITTED, READ COMMITTED, SERIALIZABLE).

Da bi se uspješno mogao obaviti primjer koji slijedi i zadaci **Zadatak 3.**, **Zadatak 4.**

i **Zadatak 5.** potrebno je za relacije mjesto i student kreirati primarne ključeve i omogućiti zaključavanje na razini n-torke. To se može postići sljedećim SQL naredbama:

```
ALTER TABLE mjesto ADD CONSTRAINT PRIMARY KEY (pbr)
                        CONSTRAINT pkMjesto;
ALTER TABLE student ADD CONSTRAINT PRIMARY KEY (JMBAG)
                        CONSTRAINT pkStudent;
```

```
ALTER TABLE mjesto LOCK MODE(ROW);
ALTER TABLE student LOCK MODE (ROW);
```

```
BEGIN WORK;
SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
SELECT * FROM mjesto WHERE pbr IN (10000, 21000, 23000);
- u ovom trenutku nema postavljenih ključeva
```

Obaviti COMMIT WORK; ili ROLLBACK WORK; jer u protivnom neće biti moguće promijeniti razinu izolacije naredbama koje slijede.

```
BEGIN WORK;
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
SELECT * FROM mjesto WHERE pbr IN (10000, 21000, 23000);
- u ovom trenutku nema postavljenih ključeva. Za vrijeme čitanja n-torki samo se provjeravalo ima li na n-torki (stranici) postavljen ekskluzivni ključ kojeg je postavila neka druga transakcija
```

```
BEGIN WORK;
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
SELECT * FROM mjesto WHERE pbr IN (10000, 21000, 23000);
```

- u ovom trenutku su pročitane n-torke (ili stranice) u relaciji mjesto zaključane ključem za čitanje i bit će otključane tek u trenutku završetka transakcije

Kod SELECT naredbi koje se izvode u interaktivnom okruženju, moguće je da će se na ekranu ispisati dio n-torki, a da će nakon toga naredba završiti u pogrešci zbog pokušaja postavljanja ključa za čitanje na n-torku (stranicu) koja je zaključana ključem za pisanje.

Zadatak 3.

Koristeći SQL Editor (Server Studio), kao korisnik *bpadmin*, uspostavite dvije korisničke sjednice s bazom podataka *studAdmin*.

U prvoj sjednici izvedite sljedeće naredbe:

```
BEGIN WORK;  
UPDATE student SET prezimeStudent = 'Novak'  
WHERE JMBAG = '0555005707';
```

a) U drugoj sjednici izvedite sljedeće naredbe:

```
BEGIN WORK;  
SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;  
SELECT * FROM student;
```

Jesu li naredbe uspješno izvedena? Obrazložite.

U drugom prozoru obavite naredbu:

```
ROLLBACK WORK;
```

b) U drugoj sjednici izvedite sljedeće naredbe:

```
BEGIN WORK;  
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;  
SELECT * FROM student;
```

Jesu li naredbe uspješno izvedena? Obrazložite.

U drugom prozoru obavite naredbu:

```
ROLLBACK WORK;
```

c) U drugoj sjednici izvedite sljedeće naredbe:

```
BEGIN WORK;  
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;  
SELECT * FROM student;
```

Jesu li naredbe uspješno izvedena? Obrazložite.

U drugom prozoru obavite naredbu:

```
ROLLBACK WORK;
```

U prvom prozoru izvedite naredbu:

```
ROLLBACK WORK;
```

Zadatak 4.

a) Korisnik *bpadmin* je jedini korisnik priključen na bazu podataka *studAdmin*. Obavlja sljedeći niz SQL naredbi:

```
{1} BEGIN WORK;  
{2} SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;  
{3} SELECT * FROM student  
      WHERE JMBAG IN ('0555000849', '0555002710');  
{4} UPDATE student SET prezimeStudent = 'Novak'  
      WHERE JMBAG IN ('0555000849', '0555002710');  
{5} COMMIT WORK;
```

Hoće li uspjeti obaviti sve naredbe?

- b) Korisnik *bpadmin* je uspostavio dvije korisničke sjednice (A i B) s bazom podataka *studAdmin*, a istovremeno nema drugih priključenih korisnika. U obje sjednice obavlja naredbe navedene u a) dijelu zadatka, ali tako da prvu naredbu izvede u *sjednici* A, pa prvu naredbu u *sjednici* B, pa zatim drugu naredbu u *sjednici* A, pa drugu naredbu u *sjednici* B, itd. Hoće li uspjeti obaviti sve naredbe?

Zadatak 5.

U zadatku se podrazumijeva korištenje IBM Informix SUBP-a. Nad bazom podataka *studAdmin* su aktivne samo dvije sjednice: A i B. Korisnici u sjednicama izvode naredbe pomoću interaktivnog alata za izvođenje SQL naredbi (npr. Server Studio): korisnik u sjednici A izvede jednu naredbu do kraja, pa korisnik u sjednici B izvede jednu naredbu do kraja, itd. Ako se jednom od korisnika zbog zaključavanja dogodi pogreška, on prestaje obavljati daljnje naredbe ali ne prekida transakciju, a drugi korisnik nastavlja s radom do završetka ili dok se i njemu ne dogodi pogreška. Za svaku naredbu (kojom mogu biti postavljeni ključevi) napišite koja vrsta ključa se postavlja na koji objekt, a ako se ključ ne postavlja obrazložite zbog čega. Ako će prilikom izvođenja naredbi doći do pogreške navedite kojem korisniku (u kojoj sjednici) pri izvođenju koje naredbe će pogreška biti dojavljena te obrazložite uzrok pogreške.

Zadatke a) i b) treba rješavati potpuno neovisno jedan od drugoga.

a)

Sjednica A	Sjednica B
<pre>{1} BEGIN WORK; {3} SET TRANSACTION ISOLATION LEVEL READ COMMITTED; {5} SELECT * FROM student; {7} UPDATE student SET prezimeStudent = 'Novak' WHERE JMBAG = '0555000235'; {9} COMMIT WORK;</pre>	<pre>{2} BEGIN WORK; {4} SET TRANSACTION ISOLATION LEVEL READ COMMITTED; {6} SELECT * FROM student WHERE JMBAG = '0555000235'; {8} UPDATE student SET prezimeStudent = 'Novak' WHERE JMBAG = '0555000256'; {10} COMMIT WORK;</pre>

b)

Sjednica A	Sjednica B
<pre>{1} BEGIN WORK; {3} SET TRANSACTION ISOLATION LEVEL SERIALIZABLE; {5} SELECT * FROM mjesto WHERE pbr IN (10000, 21000, 23000); {7} UPDATE mjesto SET nazMjesto = 'ZAGREB' WHERE pbr = 10000; {9} UPDATE mjesto SET nazMjesto = 'SPLIT' WHERE pbr = 21000; {11} COMMIT WORK;</pre>	<pre>{2} BEGIN WORK; {4} SET TRANSACTION ISOLATION LEVEL READ COMMITTED; {6} SELECT * FROM mjesto WHERE pbr = 10000; {8} UPDATE mjesto SET nazMjesto = 'SPLIT' WHERE pbr = 21000; {10} UPDATE mjesto SET nazMjesto = 'KARLOVAC' WHERE pbr = 47000; {12} COMMIT WORK;</pre>

Primarne ključeve kreirane za relacije mjesto i student moguće je ukloniti izvođenjem sljedećih SQL naredbi:

```
ALTER TABLE mjesto DROP CONSTRAINT pkMjesto;
ALTER TABLE student DROP CONSTRAINT pkStudent;
```

Zaključavanje vratiti na pretpostavljeno (default) naredbama:

```
ALTER TABLE mjesto LOCK MODE(PAGE);
ALTER TABLE student LOCK MODE (PAGE);
```

Rješenja

Zadatak 1.

- a) Pokušaj izvođenja naredbe
`DATABASE studAdmin EXCLUSIVE;`
završi greškom:
SQL Error (-425): Database is currently opened by another user.
ISAM error: Lock Timeout Expired

Budući da je u drugoj sjednici uspostavljena veza s istom bazom podataka u prvoj nije moguće bazu podataka ekskluzivno zaključati.

- b) Pokušaj izvođenja naredbe
`DATABASE studAdmin EXCLUSIVE;`
završi uspjehom:
Statement executed.
Budući da je ovo jedina aktivna korisnička sjednica s bazom podataka *studAdmin* moguće ju je ekskluzivno zaključati

Zadatak 2.

- a) Pokušaj izvođenja naredbi:
`BEGIN WORK;`
`LOCK TABLE upisanPredmet IN SHARE MODE;`
u drugoj sjednici će završiti uspješno. Dok god ne postoji process koji je tablicu zaključao ekskluzivno bilo koji proces može zaključati tablicu u dijeljenom (SHARE) modu.
- b) Pokušaj izvođenja naredbi:
`BEGIN WORK;`
`LOCK TABLE upisanPredmet IN EXCLUSIVE MODE;`
u drugoj sjednici će završiti greškom:
SQL Error (-289): Cannot lock table
(bpadmin.upisanpredmet) in requested mode.
ISAM error: Lock Timeout Expired
Budući da je u prvoj sjednici tablica *upisanPredmet* zaključana u dijeljenom modu nijedan proces je ne može zaključati ekskluzivno.

Zadatak 3.

- a) Naredbe su uspješno izvedene. Budući da je za transakciju postavljena razina izolacije `READ UNCOMMITTED`, čitanje n-torki iz relacije *student* se provodi bez zaključavanja i bez provjere jesu li zaključane.
- b) Pokušaj izvođenja naredbi završi pogreškom:
SQL Error (-244): Could not do a physical-order read to fetch next row.
ISAM error: Lock Timeout Expired

Za transakciju je postavljena razina izolacije `READ COMMITTED`, pa se provjerava je li trenutno pročitani podatak zaključan za pisanje. Time se postiže čitanje isključivo potvrđenih n-torki. Provjera se obavlja

postavljanjem „kratkotrajnog” ključa za čitanje. Budući da je n-toka s JMBAG-om '0555005707' zaključana za pisanje, pri pokušaju postavljanja ključa za čitanje na tu n-torku dolazi do pogreške.

- c) Pokušaj izvođenja naredbi završi pogreškom:
 SQL Error (-243): Could not position within a table (studadmin.student).
 ISAM error: Lock Timeout Expired

Za transakciju je postavljena razina izolacije **SERIALIZABLE**, pa se provjerava je li trenutno pročitani podatak zaključan za pisanje. Time se postiže čitanje isključivo potvrđenih n-torki. Provjera se obavlja postavljanjem ključa za čitanje koji traje do kraja transakcije. Budući da je n-toka s JMBAG-om '0555005707' zaključana za pisanje, pri pokušaju postavljanja ključa za čitanje na tu n-torku dolazi do pogreške.

Zadatak 4.

- a) Sve naredbe će se uspješno moći obaviti.
 Naredbom označenom s {3} postavlja se ključ za čitanje na n-torke relacije *student* s JMBAG-om '0555000849' ili '0555002710'.

Naredbom označenom s {4} postavlja se ključ za pisanje na n-torke s JMBAG-om '0555000849' ili '0555002710'. Budući da nema drugih sjednica s bazom *studAdmin*, postavljanje ključeva će završiti uspješno.

Uočite sljedeće: ključevi za čitanje postavljeni naredbom {3} neće spriječiti postavljanje ključa za pisanje naredbom {4} budući da se radi o ključevima koji se postavljaju u okviru iste transakcije. Drugim riječima, transakcija nikad sama sebe ne sprječava u postavljanju ključeva.

- b)

Sjednica A	Sjednica B
{1} BEGIN WORK; {3} SET TRANSACTION ISOLATION LEVEL SERIALIZABLE; {5} SELECT * FROM student WHERE JMBAG IN ('0555000849', '0555002710'); {7} UPDATE student SET prezimeStudent = 'Novak' WHERE JMBAG IN ('0555000849', '0555002710'); {9} COMMIT WORK;	{2} BEGIN WORK; {4} SET TRANSACTION ISOLATION LEVEL SERIALIZABLE; {6} SELECT * FROM student WHERE JMBAG IN ('0555000849', '0555002710'); {8} UPDATE student SET prezimeStudent = 'Novak' WHERE JMBAG IN ('0555000849', '0555002710'); {10} COMMIT WORK;
{5} postavlja se ključ za čitanje na n-torke s JMBAG-om '0555000849' ili '0555002710'. Postavljeni ključevi traju do kraja transakcije (do obavljanja naredbe {9}).	{6} postavlja se ključ za čitanje na n-torke s JMBAG-om '0555000849' ili '0555002710'. Naredba će se uspješno obaviti jer je ključ za čitanje moguće postaviti bez obzira na ključeve za čitanje postavljene u okviru sjednice A. Postavljeni ključevi traju do kraja transakcije (do obavljanja naredbe {10}).
{7} pokušaj postavljanja ključa za pisanje na n-torke s	{8} pokušaj postavljanja ključa za pisanje na n-torke s

JMBAG-om '0555000849' ili '0555002710' završava greškom jer je na te iste n-torke ključ za čitanje postavio korisnik iz sjednice B. Prestaje izvoditi daljnje naredbe. Ostaje „u transakciji“.	JMBAG-om '0555000849' ili '0555002710' završava greškom jer je na te iste n-torke ključ za čitanje postavio korisnik iz sjednice A. Prestaje izvoditi daljnje naredbe. Ostaje „u transakciji“.
---	---

Zadatak 5.

a)

Sjednica A	Sjednica B
<pre>{1} BEGIN WORK; {3} SET TRANSACTION ISOLATION LEVEL READ COMMITTED; {5} SELECT * FROM student; {7} UPDATE student SET prezimeStudent = 'Novak' WHERE JMBAG = '0555000235'; {9} COMMIT WORK;</pre>	<pre>{2} BEGIN WORK; {4} SET TRANSACTION ISOLATION LEVEL READ COMMITTED; {6} SELECT * FROM student WHERE JMBAG = '0555000235'; {8} UPDATE student SET prezimeStudent = 'Novak' WHERE JMBAG = '0555000256'; {10} COMMIT WORK;</pre>
{5} Provjerava postoji li ključ za pisanje nad svim n-torkama relacije <i>stud</i> . Zbog razine izolacije READ COMMITTED „kratkotrajni“ ključevi za čitanje po dovršetku SELECT naredbe nisu postavljeni.	{6} Provjerava postoji li ključ za pisanje nad n-torkom s JMBAG-om '0555000235'. Zbog razine izolacije READ COMMITTED, na n-torku s JMBAG-om '0555000235' „kratkotrajni“ ključ za čitanje nije postavljen po dovršetku SELECT naredbe.
{7} Postavlja se ključ za pisanje na n-torku s JMBAG-om '0555000235'. Postavljanje ključa je moguće jer nad tom n-torkom nije postavljen nikakav ključ u sjednici B. Ključ za pisanje traje do kraja transakcije (do obavljanja naredbe {9}).	{7} Postavlja se ključ za pisanje na n-torku s JMBAG-om '0555000256'. Postavljanje ključa je moguće jer nad tom n-torkom nije postavljen nikakav ključ u sjednici A. Ključ za pisanje traje do kraja transakcije (do obavljanja naredbe {10}).

b)

Sjednica A	Sjednica B
<pre>{1} BEGIN WORK; {3} SET TRANSACTION ISOLATION LEVEL SERIALIZABLE; {5} SELECT * FROM mjesto WHERE pbr IN (10000, 21000, 23000); {7} UPDATE mjesto SET nazMjesto = 'ZAGREB' WHERE pbr = 10000; {9} UPDATE mjesto SET nazMjesto = 'SPLIT' WHERE pbr = 21000; {11} COMMIT WORK;</pre>	<pre>{2} BEGIN WORK; {4} SET TRANSACTION ISOLATION LEVEL READ COMMITTED; {6} SELECT * FROM mjesto WHERE pbr = 10000; {8} UPDATE mjesto SET nazMjesto = 'SPLIT' WHERE nazMjesto = 21000; {10} UPDATE mjesto SET nazMjesto = 'KARLOVAC' WHERE pbr = 47000; {12} COMMIT WORK;</pre>
{5} Postavlja ključ za čitanje na n-torke s pbr 10000, 21000 i 23000. Postavljeni ključevi traju do kraja transakcije (do obavljanja naredbe {11}).	{6} Provjerava postoji li ključ za pisanje nad n-torkom s pbr = 10000. Zbog razine izolacije READ COMMITTED, na n-torku s pbr = 10000 „kratkotrajni“ ključ za čitanje nije postavljen po dovršetku SELECT naredbe.
{7} Pokušava postaviti ključ za pisanje na n-torku s pbr = 10000. Postavljanje ključa je moguće jer nad tom n-torkom nije postavljen nikakav ključ u sjednici B. Ključ za pisanje traje do kraja transakcije (do obavljanja naredbe {11}).	{8} Pokušava postaviti ključ za pisanje na n-torku s pbr = 21000. Postavljanje ključa nije moguće jer je nad tom n-torkom u sjednici A naredbom {5} postavljen ključ za čitanje. Prestaje izvoditi daljnje naredbe. Ostaje „u transakciji“.
{9} Pokušava postaviti ključ za pisanje na n-torku s mbr = 21000. Postavljanje ključa je moguće jer nad tom n-torkom nije postavljen nikakav ključ u sjednici B. Ključ za pisanje traje do kraja transakcije (do	

obavljanja naredbe {11})	
--------------------------	--