Laboratorijske vježbe iz Organizacije obrade podataka

PL/SQL

Zadaci:

1. Kreirana je tablica **TEMP (STUPAC1 NUMBER(4),**
STUPAC2 NUMBER(4),
STUPAC3 VARCHAR2(20))

U alatu TOAD unesite proceduru UNUTARNJA tako da kliknete 'Open a new Procedure Edit Window'. Proceduru kompajlirate sa Ctrl+Enter.

```
CREATE OR REPLACE PROCEDURE unutarnja
 (b IN OUT number)
IS
   x NUMBER:=0;
BEGIN
            FOR i IN 1..3 LOOP
                  x:=x+1;
                  b:=b+1;
                  INSERT INTO temp VALUES (x, b, 'unutarnja');
            END LOOP;
COMMIT;
END unutarnja;
```

U glavnom SQL prozoru upišite i izvršite vanjski PL/SQL blok u kojem se koriste procedura UNUTARNJA i tablica TEMP. Promotrite rezultate u tablici TEMP. Postupak: Kliknite 'Open a new SQL window'. Tu unosite vanjski blok naredbi, kao i ostale SQL naredbe. Za izvršavanje naredbe postavite kursor iza ";" i stisnite Ctrl+Enter.

```
DECLARE
            x       NUMBER:=0;
            br      NUMBER :=0;
BEGIN
            FOR i IN 1..2 LOOP
                  x:=x+1000;
                  br:=br+1;
                  INSERT INTO temp VALUES (x, br, 'Vanjska petlja');
                  unutarnja(br);
            END LOOP;
            COMMIT;
END;
```

Nakon izvršavanja ovog programskog bloka u prethodno praznoj tablici TEMP nalazi se 8 redaka:

1000	1	Vanjska petlja
1	2	unutarnja
2	3	unutarnja
3	4	unutarnja
2000	5	Vanjska petlja
1	6	unutarnja
2	7	unutarnja
3	8	unutarnja

Napišite zatim **funkciju KVADRIRAJ** koja će vraćati kvadrat ulaznog parametra. U već postojeću proceduru UNUTARNJA unesite promjene tako da varijabla x unutar petlje pri svakom prolazu poprima vrijednost kvadrata varijable b.
Za promjenu **procedure** UNUTARNJA u Schema Browseru izaberite dotičnu proceduru, te desnim klikom odaberite 'Load in Procedure Editor'.

Funkcija KVADRIRAJ ima sljedeći izgled:

```
CREATE OR REPLACE FUNCTION kvadriraj
       (y IN NUMBER)
RETURN NUMBER
IS
BEGIN
      RETURN y*y;
END kvadriraj;
```

Izmijenjena prodedura UNUTARNJA izgleda ovako:

```
procedure unutarnja
      (b IN OUT number)
IS
      x NUMBER:=0;
BEGIN
      FOR i IN 1..3 LOOP
            x:=kvadriraj(b);
            b:=b+1;
            INSERT INTO temp VALUES (x,b,'unutarnja');
      END LOOP
COMMIT;
END unutarnja;
```

Izvršavanjem ranije prikazanog programskog bloka, u tablicu TEMP dodaje se još 8 redaka. Prikazan je svih 16 redaka u tablici. Posljednje uneseni reci nalaze se na dnu:

1000	1	Vanjska petlja
1	2	unutarnja
2	3	unutarnja
3	4	unutarnja
2000	5	Vanjska petlja
1	6	unutarnja
2	7	unutarnja
3	8	unutarnja
1000	1	Vanjska petlja
1	2	unutarnja
4	3	unutarnja
9	4	unutarnja
2000	5	Vanjska petlja
25	6	unutarnja
36	7	unutarnja
49	8	unutarnja

2. Kreirana je tablica **STATUS** koja ima samo jedan tekstualni stupac s 50 znakova, te tablice **EMPLOYEE** i **DEPARTMENT**.

   Kreirajte proceduru **INSERT_EMPLOYEE** koja će ubacivati slogove u tablicu EMPLOYEE. **Ulazni parametri** bit će ***(e_id number, e_name varchar2, e_job varchar2, mgr_id varchar2, hire_date date, sal number, dept_id number)***. Uspješnost upisa evidentira se u tablici STATUS.

   Greške pri upisu unutar procedura hvataju se preko iznimaka (EXCEPTION dio). Upišite razlog pogreške u tablicu STATUS. Predvidite tri moguće pogreške:

   A) unos *stringa* umjesto broja (hvata se iznimkom INVALID_NUMBER),
      obratite pažnju na parametar **mgr_id** procedure INSERT_EMPLOYEE koji je varchar2. U tablici EMPLOYEE manager_id je tipa number. Prilikom unosa sustav pokušava pretvoriti niz znakova u broj. Za '23' unos je uspješan, a za '2A' baca se iznimka INVALID_NUMBER.
   B) povredu primarnog ključa,
   C) povredu referencijalnog integriteta.

   Iznimke B) i C) uhvatite preko OTHERS, a onda (unutar obrade iznimke) preko IF-THEN blokova utvrdite da li je došlo do povrede referencijalnog integriteta, odnosno primarnog ključa.
      (Naputak: Stvorite dvije lokalne varijable koje će provjeravati da li u tablici EMPLOYEE već postoji *id* zaposlenika kojeg želimo unijeti, odnosno da li u tablici DEPARTMENT postoji broj odjela dotičnog zaposlenika. Koristite upit oblika `Select count(*) INTO ... FROM ... )`

Isprobajte funkciju sa par točnih, odnosno par pogrešnih upisa i pogledajte rezultate u tablici STATUS.

Za upis koristite sljedeću skriptu u SQL Prozoru

```
DECLARE
  E_ID NUMBER;
  E_NAME VARCHAR2(200);
  E_JOB VARCHAR2(200);
  MGR_ID VARCHAR2(200);
  HIRE_DATE DATE;
  SAL NUMBER;
  DEPT_ID NUMBER;

BEGIN
  E_ID := 1;
  E_NAME := 'ime';
  E_JOB := 'prezime';
  MGR_ID := '23';
  HIRE_DATE := to_date('23.05.2002','DD.MM.YYYY');
  SAL := 233;
  DEPT_ID := 10;

  INSERT_EMPLOYEE ( E_ID, E_NAME, E_JOB, MGR_ID, HIRE_DATE,
SAL, DEPT_ID );
  COMMIT;
END;
```

U nastavku je navedena cijela procedura INSERT_EMPLOYEE:

```
CREATE OR REPLACE PROCEDURE INSERT_EMPLOYEE
           (e_id IN NUMBER,
          e_name IN VARCHAR2,
            e_job IN VARCHAR2,
            mgr_id IN VARCHAR2,
            hire_date IN DATE,
            sal IN NUMBER,
            dept_id IN NUMBER)
AS

            counte NUMBER(4);
            countd NUMBER(4);

BEGIN

       INSERT INTO EMPLOYEE VALUES (e_id, e_name, e_job, mgr_id,hire_date,
                                                          sal, dept_id);
       INSERT INTO STATUS VALUES ('Uspjesan upis.');

       COMMIT;

EXCEPTION

       WHEN INVALID_NUMBER THEN
              ROLLBACK;
              INSERT INTO STATUS VALUES ('Pokusaj unosa Stringa kao
                                                          broja.');
              COMMIT;
      WHEN others THEN
              ROLLBACK;
              SELECT COUNT(*) INTO counte FROM EMPLOYEE WHERE id=e_id;
              IF counte=1 THEN
```

```
                INSERT INTO STATUS VALUES ('Povreda primarnog kljuca.');
            ELSE
                SELECT COUNT(*) INTO countd FROM DEPARTMENT WHERE
                                                          id=dept_id;
                IF countd=0 THEN
                INSERT INTO status VALUES ('Povreda referencijalnog
                                                       integriteta.');
                END IF;
            END IF;
            COMMIT;

END INSERT_EMPLOYEE;
```

3. Kreirana je procedura TRANSAKCIJE sljedećeg sadržaja:

```
                CREATE OR REPLACE procedure transakcije
                IS
(1)             BEGIN
                  --priprema
(2)               DELETE FROM STATUS;
(3)               COMMIT;
                  --završetak pripreme
(4)               INSERT INTO STATUS VALUES ('Unos br. 1');
(5)               SAVEPOINT save_a;
(6)               INSERT INTO STATUS VALUES ('Unos br. 2');
(7)               SAVEPOINT save_b;
(8)               DELETE FROM STATUS WHERE status = 'Unos br. 1';
(9)               SAVEPOINT save_c;
(10)              INSERT INTO STATUS VALUES ('Unos br. 3');
(11)              ROLLBACK TO SAVEPOINT save_c;
(12)              INSERT INTO STATUS VALUES ('Unos br. 4');
(13)              ROLLBACK TO SAVEPOINT save_a;
(14)              COMMIT;
(15)            END;
```

Pokrenite proceduru:

```
BEGIN
  transakcije;
END;
```

Napomena: linije su označene brojevima od 1 do 15 radi boljeg razumijevanja i ne upisuju se u prozor za unos procedure.

Prije nego što pokrenete ovu proceduru odgovorite (za sebe) na sljedeća pitanja:

a) Koje su naredbe za transakcijsku obradu?
b) Što radi naredba COMMIT?
c) Čemu služi SAVEPOINT naredba?
d) Što će se nalaziti zapisano u tablici STATUS nakon izvršenja svake od naredbi u PL/SQL bloku?
e) Što će na kraju biti zapisano u tablici STATUS?

Izvršite proceduru TRANSAKCIJE i provjerite rezultat.

Koja će transakcije ostati trajno zapisane kao rezultat izvršavanja procedure ukoliko redoslijed izmijenimo tako da bude sljedeći:

f) (1)-(2)-(3)-(4)-(6)-(8)-(10)-(12)-(14)-(15)

g) (1)-(2)-(3)-(4)-(6)-(7)-(8)-(9)-(10)-(11)-(12)-(14)-(15)
h) (1)-(2)-(3)-(4)-(5)-(6)-(7)-(8)-(9)-(10)-(13)-(12)-(11)-(14)-(15)
i) (1)-(2)-(3)-(4)-(5)-(6)-(7)-(8)-(9)-(10)-(13)-(12)-(14)-(15)

Odgovori na pitanja d) e) f) g) h) i):

d)

- 3 – prazna tablica
- 4 – `Unos br. 1`,
- 6 – `Unos br. 1, Unos br. 2`,
- 8 – `Unos br. 2`,
- 10 – `Unos br. 2, Unos br. 3`,
- 11 – `Unos br. 2` ,
- 10 – `Unos br. 2, Unos br. 4`,
- 11 – `Unos br. 1`

e)

- `Unos br. 1`

f)

- `Unos br. 2, Unos br. 3, Unos br. 4`

g)

- `Unos br. 2, Unos br. 3, Unos br. 4`

h)

- *Sustav javlja grešku: savepoint 'SAVE_C' never established budu•i da do trenutka postavljanja Savepoint SAVE_A (naredba 5) nije postavljen Savepoint SAVE_C*

i)

- `Unos br. 1, Unos br. 4`