Adatbázisok 1, 3. ZH, 2020.05.13 szerda, 16.05

==============================================

Az alábbi PL/SQL feladatokat kell megoldani. Csak helyes mûködésû programot értékelek,

részpontszám nem szerezhetõ a feladatokra.

2 feladat megoldása elég a 2-es jegyhez,

3 feladat megoldása a 3-as jegyhez,

4 feladat megoldása a 4-es jegyhez,

5 feladat megoldása az 5-ös jegyhez.

A procedúrákat és függvényeket olyan névvel és paraméterezéssel kell létrehozni, ahogyan

alább szerepel. A programokat a megadott paraméterekkel meg is kell hívni és az eredményt

és a forráskódot kell emailben elküldeni a nikovits@inf.elte.hu email címre.

1. feladat

----------

Írjunk meg egy függvényt, amelyik a paraméterül kapott két számnak visszaadja

a legkisebb közös többszörösét.

CREATE OR REPLACE FUNCTION lkkt(p1 integer, p2 integer) RETURN number IS

Tesztelés:

SELECT lkkt(33462, 18876) FROM dual;

----------------------------------------------------------------------------

CREATE OR REPLACE FUNCTION lkkt(p1 integer, p2 integer) RETURN number IS

lkkt number;

n number;

BEGIN

n := p1 \* p2;

FOR i IN 1..n LOOP

IF mod(i,p1) = 0 AND mod(i,p2) = 0 THEN

lkkt := i;

return lkkt;

END IF;

END LOOP;

return n;

END;

/

----------------------------------------------------------------------------

SELECT lkkt(33462, 18876) FROM dual;

----------------------------------------------------------------------------

EREDMÉNY:

736164

----------------------------------------------------------------------------

2. feladat (Elõtte készítsenek saját példányt a nikovits.dolgozo táblából.)

----------

Írjunk meg egy procedúrát, amelyik megnöveli azon dolgozók fizetését, akik a paraméterként

megadott fizetési kategóriába esnek. A fizetést a dolgozó jutalékával kell megnövelni,

illetve, ha a dolgozó jutaléka NULL érték, akkor 100-zal.

A procedúra a módosítás után írja ki az összes dolgozó átlagfizetését (az új fizetések átlagát).

A procedúra a futása végén adjon ki egy ROLLBACK utasítást, azért, hogy az adatok

ne változzanak meg, és legközelebb futtatva is megfelelõ eredményt írjon ki.

CREATE OR REPLACE PROCEDURE add\_jutalek(p integer) IS

Tesztelés:

set serveroutput on

execute add\_jutalek(4);

----------------------------------------------------------------------------

SEGÉD FÜGGVÉNY:

CREATE OR REPLACE FUNCTION jut(id integer) RETURN integer IS

penz integer;

BEGIN

SELECT COALESCE(jutalek,100) INTO penz FROM nikovits.dolgozo WHERE dkod = id;

return penz;

END;

/

----------------------------------------------------------------------------

CREATE OR REPLACE PROCEDURE add\_jutalek(p integer) IS

atlag number;

BEGIN

UPDATE dolgozo2

SET fizetes = fizetes + jut(dkod)

WHERE dkod IN

(SELECT D.dkod FROM NIKOVITS.dolgozo D, NIKOVITS.fiz\_kategoria F

WHERE D.fizetes BETWEEN F.also AND F.felso AND F.kategoria = p);

SELECT ROUND(AVG(fizetes)) INTO atlag FROM dolgozo2;

DBMS\_OUTPUT.PUT\_LINE(atlag);

ROLLBACK;

END;

/

----------------------------------------------------------------------------

set serveroutput on

execute add\_jutalek(4);

----------------------------------------------------------------------------

EREDMÉNY:

2231

----------------------------------------------------------------------------

3. Feladat

----------

Írjunk meg egy plsql procedúrát, amelyik veszi a dolgozókat ábácé szerinti sorrendben,

és minden páros sorszámú dolgozó nevét és fizetését beleteszi egy asszociatív tömbbe.

A procedúra a végén írja ki a tömb utolsó két sorát.

CREATE OR REPLACE PROCEDURE proc3 IS

Tesztelés:

set serveroutput on

execute proc3();

----------------------------------------------------------------------------

CREATE OR REPLACE PROCEDURE proc3 IS

CURSOR curs IS SELECT dnev, fizetes FROM nikovits.dolgozo ORDER BY dnev;

dolg curs%ROWTYPE;

TYPE tab\_type IS TABLE OF curs%ROWTYPE INDEX BY BINARY\_INTEGER;

tab tab\_type;

aktszam integer;

darab integer;

BEGIN

aktszam := 0;

darab := 0;

OPEN curs;

LOOP

aktszam := aktszam + 1;

FETCH curs into dolg;

EXIT WHEN curs%NOTFOUND;

IF mod(aktszam,2) != 1 THEN

darab := darab + 1;

tab(darab) := dolg;

END IF;

END LOOP;

CLOSE curs;

DBMS\_OUTPUT.PUT\_LINE(tab(darab - 1).dnev || ' ' || tab(darab - 1).fizetes);

DBMS\_OUTPUT.PUT\_LINE(tab(darab).dnev || ' ' || tab(darab).fizetes);

END;

/

----------------------------------------------------------------------------

set serveroutput on

execute proc3();

----------------------------------------------------------------------------

EREDMÉNY:

SMITH 800

WARD 1250

----------------------------------------------------------------------------

4. feladat (Elõtte készítsenek saját példányt a nikovits.dolgozo táblából.)

----------

Írjunk meg egy procedúrát, amelyik a paraméterében megadott azonosítójú osztályon

módosítja a fizetéseket a következõk szerint: ha a dolgozó nevében van 'T' betû, akkor

növelje meg a fizetést 10000-rel. Ha nincs, akkor annyiszor 10000-rel növelje meg,

amennyi a dolgozó fizetési kategóriája. A procedúra írja ki soronként a dolgozó

nevét és módosított (új) fizetését.

A procedúra a futása végén adjon ki egy ROLLBACK utasítást, azért, hogy az adatok

ne változzanak meg, és legközelebb futtatva is megfelelõ eredményt írjon ki.

CREATE OR REPLACE PROCEDURE fiz\_novel(p\_oazon number) IS

Tesztelés:

set serveroutput on

execute fiz\_novel(20);

5. Feladat

----------

Írjunk meg egy procedúrát, amelyik a NIKOVITS.VAGYONOK tábla alapján kiírja azoknak

a személyeknek a nevét, vagyonát, valamint leszármazottainak átlagos vagyonát, akikre igaz,

hogy a leszármazottainak átlagos vagyona nagyobb, mint az illetõ vagyona.

A program tehát soronként 3 adatot ír ki: név, vagyon, leszármazottak átlagos vagyona

CREATE OR REPLACE PROCEDURE gazdag\_leszarmazottak IS

Tesztelés:

set serveroutput on

execute gazdag\_leszarmazottak();