Adatbázisok 1, 3. ZH, 2020.05.13 szerda, 12.15

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

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 számnak visszaadja

a prímtényezõs felbontását egy karakterláncban.

Például: 120-ra visszaadja, hogy 2\*2\*2\*3\*5.

CREATE OR REPLACE FUNCTION felbontas(p1 integer) RETURN varchar2 IS

Tesztelés:

SELECT felbontas(86632) FROM dual;

CREATE OR REPLACE FUNCTION prim(n integer) RETURN boolean IS

is\_prime boolean;

begin

if n < 2 then

is\_prime := false;

return is\_prime;

end if;

is\_prime := true;

for i in 2..sqrt(n) loop

if n mod i = 0 then

is\_prime := false;

return is\_prime;

end if;

end loop;

return is\_prime;

end;

/

CREATE OR REPLACE FUNCTION felbontas(p1 integer) RETURN varchar2 IS

p2 integer := p1;

outchar varchar2(32000) := '';

begin

if prim(p1) or p1 = 0 or p1 = 1 then

return to\_char(p1);

else

for i in 2..p2-1 loop

while p2 mod i = 0 loop

outchar := outchar || to\_char(i) || '\*';

p2 := p2/i;

end loop;

end loop;

end if;

return trim(trailing '\*' from outchar);

end;

/

SELECT felbontas(86632) FROM dual;

/\*

Kimenet:

2\*2\*2\*7\*7\*13\*17

\*/

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

----------

Írjunk meg egy procedúrát, amelyik megnöveli a dolgozók fizetését a fõnökük fizetésével.

Akinek nincs fönöke, annak ne változzon a fizetése.

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\_fonokFiz IS

Tesztelés:

set serveroutput on

execute add\_fonokFiz();

CREATE OR REPLACE PROCEDURE add\_fonokFiz IS

cursor curs is

(select d1.dnev fonok, d1.fizetes fonokf, d2.dnev beo, d2.fizetes beof

from dolgozo2 d1, dolgozo2 d2

where d1.dkod = d2.fonoke);

rec curs%rowtype;

n number;

avgf number;

begin

open curs;

loop

fetch curs into rec;

exit when curs%NOTFOUND;

n := rec.fonokf;

update dolgozo2

set fizetes = fizetes + n

where dnev = rec.beo;

end loop;

close curs;

select avg(fizetes) into avgf from dolgozo2;

DBMS\_OUTPUT.PUT\_LINE('Az új fizetések átlaga: ' || avgf);

rollback;

end;

/

set serveroutput on

execute add\_fonokFiz();

/\*

Kimenet:

5886,764705882352941176470588235294117647

\*/

3. feladat

----------

Írjunk meg egy procedúrát, amelyik veszi az elsõ n prímszámot (1. prímszám: 2, 2. prímszám: 3, stb.)

és beleteszi azokat egy asszociatív tömbbe. A procedúra a végén írja ki a tömb utolsó elemét és a

prímszámok összegét.

\*/

CREATE OR REPLACE PROCEDURE primes(n integer) IS

...

set serveroutput on

execute primes(100);

CREATE OR REPLACE FUNCTION prim(n integer) RETURN boolean IS

is\_prime boolean;

begin

if n < 2 then

is\_prime := false;

return is\_prime;

end if;

is\_prime := true;

for i in 2..sqrt(n) loop

if n mod i = 0 then

is\_prime := false;

return is\_prime;

end if;

end loop;

return is\_prime;

end;

/

CREATE OR REPLACE PROCEDURE primes(n integer) IS

type table\_type is table of integer index by binary\_integer;

primes table\_type;

darab integer;

szam integer;

osszeg integer := 0;

begin

darab := 0;

szam := 2;

while darab < n loop

if prim(szam) then

darab := darab + 1;

primes(darab) := szam;

end if;

szam := szam + 1;

end loop;

for i in 1..n loop

osszeg := osszeg + primes(i);

end loop;

DBMS\_OUTPUT.PUT\_LINE('Az utolsó prímszám: ' || primes(n));

DBMS\_OUTPUT.PUT\_LINE('A prímszámok összege: ' || osszeg);

end;

/

set serveroutput on

execute primes(100);

/\*

Kimenet:

Az utolsó prímszám: 541

A prímszámok összege: 24133

\*/

4. Feladat

----------

Írjunk meg egy függvényt, amelyik visszaadja a paraméterként szereplõ '+'-szal

elválasztott számok és kifejezések összegét. Ha valamelyik kifejezés nem szám,

akkor azt az összeadásnál hagyja figyelmen kívül, vagyis 0-nak tekintse.

CREATE OR REPLACE FUNCTION osszeg2(p\_char VARCHAR2) RETURN number IS

Tesztelés:

SELECT osszeg2('1+21 + bubu + 1y2 + 2 + -1 + 0 +') FROM dual;

CREATE OR REPLACE FUNCTION osszeg2(p\_char VARCHAR2) RETURN number IS

osszeg NUMBER := 0;

temp VARCHAR2(1000);

i INTEGER;

FUNCTION is\_number(param\_char VARCHAR2) RETURN number IS

BEGIN

IF param\_char is null THEN

return 0;

END IF;

return to\_number(param\_char);

EXCEPTION

WHEN OTHERS THEN RETURN 0;

END;

BEGIN

temp := replace(p\_char, ' ', '');

i := instr(temp, '+') - 1;

WHILE length(temp) > 0 LOOP

osszeg := osszeg + is\_number(substr(temp, 0,i));

temp := substr(temp, i+2);

i := instr(temp, '+') - 1;

END LOOP;

RETURN osszeg;

END;

/

SELECT osszeg2('1+21 + bubu + 1y2 + 2 + -1 + 0 +') FROM dual;

/\*

Kimenet:

23

\*/

5. feladat

----------

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

a személyeknek a nevét és vagyonát, akikre igaz, hogy sem az õ nevük, sem egyetlen

felmenõjük neve sem tartalmazza a paraméterként megadott karakterláncot.

CREATE OR REPLACE PROCEDURE kar\_nincs(p varchar2) IS

Tesztelés:

set serveroutput on

execute kar\_nincs('B');

CREATE OR REPLACE PROCEDURE kar\_nincs(p varchar2) IS

CURSOR curs IS

SELECT DISTINCT chnev, chvagyon

FROM

(SELECT DISTINCT parents.nev pnev, children.nev chnev, parents.vagyon pvagyon, children.vagyon chvagyon

FROM nikovits.vagyonok parents, nikovits.vagyonok children

WHERE LEVEL >= 1 AND parents.nev = children.apja

START WITH parents.nev = 'ADAM'

CONNECT BY PRIOR parents.nev = parents.apja)

WHERE pnev NOT LIKE '%'||p||'%' AND chnev NOT LIKE '%'||p||'%';

rec curs%rowtype;

BEGIN

OPEN curs;

LOOP

FETCH curs INTO rec;

EXIT WHEN curs%notfound;

DBMS\_OUTPUT.PUT\_LINE(rec.chnev||' '||rec.chvagyon);

END LOOP;

CLOSE curs;

END;

/

set serveroutput on

execute kar\_nincs('B');

/\*

Kimenet:

BALAZS 180000

CSABA 163000

KAIN 250000

PAL 133000

BARNA 340000

SAMU 176000

KAROLY 138000

\*/