Adatbázisok 2. zh1, 2022.10.25. 10.15

**Név: Tóth Botond**

**Neptun kód: MQH41V**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Fontos információ \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ELTE HKR. 377/A. §

(1) Az a hallgató, aki olyan tanulmányi teljesítménymérés (vizsga, zárthelyi,

beadandó feladat) során, amelynek keretében számítógépes program vagy programmodul

elkészítése a feladat, az oktató által meghatározottakon kívül más segédeszközt vesz igénybe,

illetve más hallgatónak meg nem engedett segítséget nyújt, tanulmányi szabálytalanságot

követ el, ezért az adott félévben a tantárgyat nem teljesítheti és a tantárgy kreditjét nem

szerezheti meg.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

A megoldásokat kérem, ebbe az állományba a feladatok után írják bele, majd

csatolva küldjék el a nikovits@inf.elte.hu email címre.

A papíros részből és a számítógépes részből is minimum 8 pont megszerzése

minimális követlemény az 1-estől különböző jegyhez.

A ponthatárok a következők lesznek: 21, 30, 39, 48.

"Papíron" megoldandó feladatok

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

1. feladat (6 pont)

Adott az alábbi B+ fa.

Tegyük fel, hogy egy B+ fa blokkjaiba 3 kulcs fér el plusz 4 mutató. A kulcsok

különbözőek. Szúrjuk be a B+ fába az alábbi kulcsértékeket a megadott sorrendben:

22, 75, 42

Adjuk meg a B+ fa minden olyan állapotát, amikor egy csomópont kettéosztására volt szükség.

71

41|59 81

15|32|36 41|44|52 59|68 71|74|78 81|87

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

**Megoldás:**

71

32|41|59 81

15|22 32|36 41|44|52 59|68 71|74|78 81|87

71

32|41|59 75|81

15|22 32|36 41|44|52 59|68 71|74 75|78 81|87

44|71

32|41 59 75|81

15|22 32|36 41|42 44|52 59|68 71|74 75|78 81|87

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

2. feladat (6 pont)

a) Tömörítsük a következő bitvektort a szakaszhossz kódolással.

000000000000001000000000011

b) Fejtsük vissza a következő, szakaszhossz kódolással tömörített bitvektort:

001110101101110111

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

**Megoldás:**

**a**

14, 10, 0

binárisan:1110 ,1010, 0

unáris előtagok: 1110, 1110, 0

Megoldás: **111011101110101000**

**b**

001110101101110111

0\_0 => 0

1110\_1011 => 11

0\_1 => 1

110\_111 => 7

Megoldás: **10000000000010100000001**

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

3. feladat (8 pont)

Lineáris hasító index felépítése

Tegyük fel, hogy egy blokkba 2 rekord fér el és a kosarak az alábbi rekordokat tartalmazzák:

1000 1111

---- ----

1100 0001 1010 1011

0000 1001 0010 0011

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

00 01 10 11

Az előre megadott küszöbszám (rekordok száma/kosarak száma) legyen 2,9.

Jelenleg n = 4 (a kosarak száma), i = 2 (bitek száma).

Szúrjuk be az alábbi hasító értékkel rendelkező sorokat egymás után, és minden újabb

kosár létrehozása után rajzoljuk le az adatszerkezetet.

0100, 1110, 0111, 0110

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

**Megoldás:**

0100 => 11/ 4 => 2,75 tehát nem kell új kosár

0100

1000 1111

---- ----

1100 0001 1010 1011

0000 1001 0010 0011

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

00 01 10 11

1110 => 12/ 4 => 3 új kosarat nyitunk

1111

----

1000 0001 1010 1011 1100

0000 1001 0010 0011 0100

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

000 001 010 011 100

0111=> 13/ 5 => 2,6 tehát nem kell új kosár

0111

1111

----

1000 0001 1010 1011 1100

0000 1001 0010 0011 0100

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

000 001 010 011 100

0110 => 14/ 5 => 2,8 tehát nem kell új kosár

0111

0110 1111

----- ----

1000 0001 1010 1011 1100

0000 1001 0010 0011 0100

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

000 001 010 011 100

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

4. feladat (10 pont)

Tekintsük a következő paraméterekkel rendelkező relációt:

T(R) = 1200000, V(R,A) = 100, V(R,B) = 500, bf(R)=50

a) Számoljuk ki T(W)-t az egyenletességi feltételezéssel élve, ha W a következő lekérdezés eredménye:

W <-- SELECT \* FROM R WHERE A=x OR B=y; (ki kell számolni, a képlet nem elég)

b) Számoljuk ki B(I)-t, ha I egy R-re létrehozott B+ fa index, amelyre bf(I) = 80?

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

**Megoldás:**

**a**

B(R) = T/bf = 24 000

T(W) = T(R)\*(1-[1-1/V(R,A)]\*[1-1/V(R,B)]) =

T(W) = T(R)/V(R,A) + T(R)/V(R,B) - T(R)/[V(R,A)\*V(R,B)] = 12 000 + 2 400 – 24 = 14 376

T(W) = **14 376**

**b**

bf(I) = 80

levélszint: T(R) / bf(I) = 15000

2.szint: 15000/bf(I) = 187,5 => 188

3.szint: 188/bf(I) = 2,35 => 3

4. szint: 1

Tehát 15000 + 188 + 3 + 1 = 15192

B(I) = **15192**

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

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

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

Számítógépes, SQL, illetve PL/SQL-ben megoldandó feladatok.

Ezeknél a feladatoknál a futtatás eredményét is kérem bemásolni a fájlba.

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

5. feladat (10 pont)

Adjuk meg a NIKOVITS felhasználó tulajdonában levő hash clusterek nevét és méretét bájtban. (Név, Méret)

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

**Megoldás:**

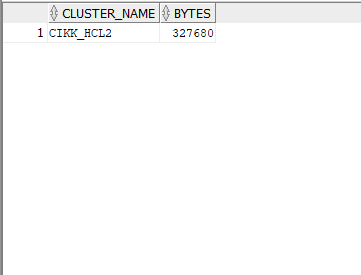
SELECT hash.cluster\_name, seg.bytes

FROM dba\_cluster\_hash\_expressions hash

JOIN dba\_objects obj ON obj.object\_name = hash.cluster\_name

JOIN dba\_segments seg ON seg.segment\_name = obj.object\_name

WHERE hash.owner = 'NIKOVITS' and obj.owner='NIKOVITS' and seg.owner='NIKOVITS';



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

6. feladat (10 pont)

Írjunk meg egy PL/SQL függvényt, amelyik egy karakterláncban visszaadja a NIKOVITS felhasználó olyan tábláinak

nevét (ábécé sorrendben, vesszővel elválasztva), amelyeknek a 2. és 5. oszlopa ugyanolyan típusú.

A hossz és pontosság nem számít, az alaptípus legyen azonos.

CREATE OR REPLACE FUNCTION tab\_2\_5 RETURN VARCHAR2 IS

...

SELECT tab\_2\_5() from dual;

Ellenőrzési lehetőség:

set serveroutput on

EXECUTE check\_plsql('tab\_2\_5()');

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

**Megoldás:**

create or replace FUNCTION tab\_2\_5 RETURN VARCHAR2 IS

CURSOR similar\_tables IS

SELECT table\_name FROM

((SELECT table\_name,data\_type

FROM dba\_tab\_columns

WHERE OWNER='NIKOVITS' and column\_id = 2)

INTERSECT

(SELECT table\_name,data\_type

FROM dba\_tab\_columns

WHERE OWNER='NIKOVITS' and column\_id = 5))

ORDER BY table\_name;

solution varchar(1000);

BEGIN

solution := '';

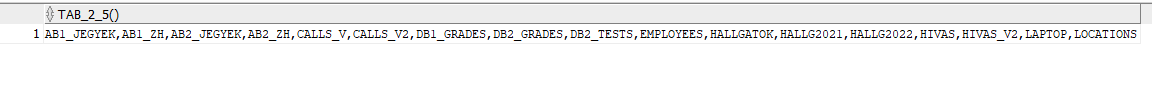
FOR table\_record IN similar\_tables LOOP

solution := solution || table\_record.table\_name || ',';

END LOOP;

return SUBSTR(solution, 1, LENGTH(solution) - 1);

END;



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

7. feladat (10 pont)

Írjunk meg egy PL/SQL procedúrát, amelyik kiírja, hogy a NIKOVITS.CUSTOMERS táblának átlagosan hány sora

van egy blokkban, vagyis a (sorok száma)/(nemüres blokkok száma) értéket egy tizedesre kerekítve. (Átlag)

CREATE OR REPLACE PROCEDURE bl\_factor IS

...

set serveroutput on

EXECUTE bl\_factor();

Ellenőrzési lehetőség:

EXECUTE check\_plsql('bl\_factor()');

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

**Megoldás:**

create or replace PROCEDURE bl\_factor IS

rows integer := 0;

blocks integer := 0;

rate float := 0;

BEGIN

SELECT COUNT(\*) INTO rows

FROM nikovits.customers;

SELECT COUNT(\*) INTO blocks

FROM

(SELECT dbms\_rowid.rowid\_block\_number(ROWID), dbms\_rowid.rowid\_relative\_fno(ROWID)

FROM NIKOVITS.customers

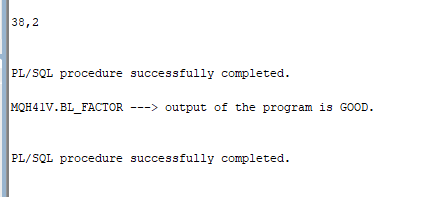
GROUP BY dbms\_rowid.rowid\_block\_number(ROWID), dbms\_rowid.rowid\_relative\_fno(ROWID));

rate := rows/blocks;

rate := ROUND(rate ,2 );

dbms\_output.put\_line(rate);

END;



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