**AB2 zh2 2022. december 06. 10:15 Név: Tóth Botond**

**Neptun kód: MQH41V**

A feladatokra összesen 60 pont szerezhető, a ponthatárok azonosak az előző zh ponthatáraival.

**„Papíron” megoldandó feladatok**

A megoldásokat ebbe az állományba írják bele, és küldjék el a [nikovits@inf.elte.hu](mailto:nikovits@inf.elte.hu) címre!

**1. feladat (5 + 5 pont)**

**a)** A következő **REDO** típusú naplóbejegyzés-sorozat a T, U és V tranzakciókra vonatkozik:

<start T> <T,A,15> <start U> <U,B,25> <T,A,35> <start V> <U,D,85> <T,C,55>   
<U,B,45> <V,E,55> <U,D,55> <COMMIT V> <COMMIT U> <T,A,55> <END V> **<T,C,80>**

Adjuk meg a helyreállításhoz szükséges tevékenységeket, ha az utolsó lemezre került naplóbejegyzés: **<T, C, 80>**. WRITE, OUTPUT, <napló>, FLUSH LOG műveletek megfelelő sorrendjét kell megadni.

**A megoldást kérem ide írják:**

WRITE(B,25)

OUTPUT(B)

WRITE(D,85)

OUTPUT(D)

WRITE(B,45)

OUTPUT(B)

WRITE(D,55)

OUTPUT(D)

<END U>

FLUSH LOG

**b)** A következő **UNDO** típusú naplóbejegyzés-sorozat a T, U és V tranzakciókra vonatkozik:

<start T> <T,E,50> <abort T> <start U> <U,A,12> <start V> <V,B,20> <U,A,40> <V,D,80> <U,C,52> <V,B,40> <V,D,50> <COMMIT V> <U,A,30> **<U,C,70>**

Adjuk meg a helyreállításhoz szükséges tevékenységeket, ha az utolsó lemezre került naplóbejegyzés: **<U, C, 70>**. WRITE, OUTPUT, <napló>, FLUSH LOG műveletek megfelelő sorrendjét kell megadni.

**A megoldást kérem ide írják:**

WRITE(C,70)

OUTPUT(C)

WRITE(A,30)

OUTPUT(A)

WRITE(C,52)

OUTPUT(C)

WRITE(A,40)

OUTPUT(A)

WRITE(A,12)

OUTPUT(A)

<ABORT U>

FLUSH LOG

**2. feladat (10 pont)**

Adjuk meg a konfliktus-sorbarendezhető ütemezések számát az alábbi tranzakció párokra.

Indokoljuk is meg az eredményt röviden!

a) **T1**: W1(A); R1(B); W1(B) **T2**: R2(C); W2(C); W2(D)

b) **T1**: W1(A); R1(B); W1(B); R1(C) **T2**: R2(B); W2(B); R2(A)

c) **T1**: R1(A); W1(A); W1(C); R1(B) **T2**: R2(A); W2(B); W2(A)

**A megoldást kérem ide írják:**

**Az aláhúzottal jelzett rész felcserélhető:**

**a)** W1(A); R1(B); W1(B); R2(C); W2(C); W2(D)

Itt a két tranzakció minden R, W párja felcserélhető mivel külön adatelemekre vonatkoznak

6!/(3!\*3!) =20

**b)** W1(A); R1(B); W1(B); R1(C)**;** R2(B); W2(B); R2(A)

W2(B) nem lehet hamarabb mint W1(B)

4!/(3!\*1!) =4

**c)** R1(A); W1(A); W1(C); R1(B);R2(A); W2(B); W2(A)

W1(A) nem lehet később mint R2(A) és R1(B) nem lehet később mint W2(B)

3!/(2!\*1!)=3

**3. feladat (4 + 4 pont)**

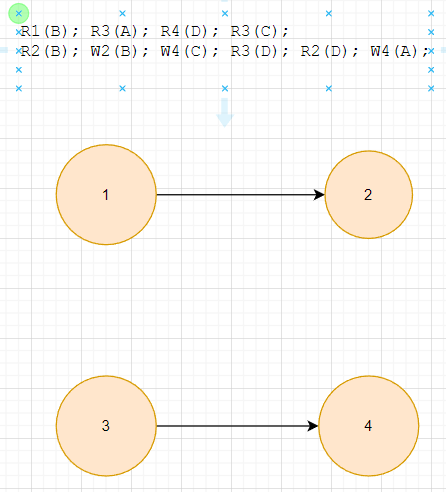
Adja meg az alábbi ütemezések megelőzési gráfját. (A gráf megrajzolásához használhatja a <https://app.diagrams.net/> programot és a honlapon szereplő **graf\_rajzolas.svg** állományt.) Adja meg, hogy hány olyan soros ütemezés van, amely konfliktus-elvivalens a megadott ütemezéssel.

**a)** R1(B); R3(A); R4(D); R3(C); R2(B); W2(B); W4(C); R3(D); R2(D); W4(A);

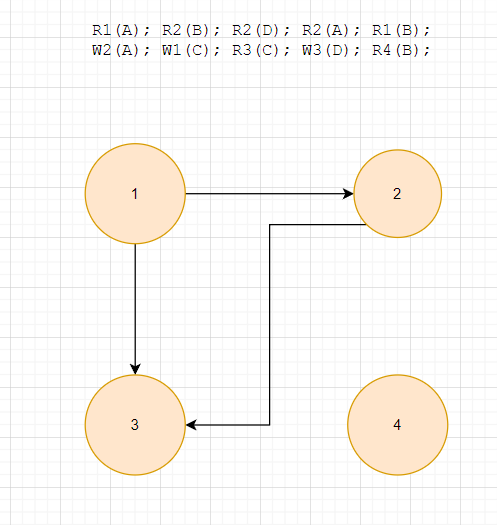
**b)** R1(A); R2(B); R2(D); R2(A); R1(B); W2(A); W1(C); R3(C); W3(D); R4(B);

**A megoldást kérem ide írják:**

1. **1234,3412,1324,1342, 3124,3142 Tehát 6 db soros ütemezés van amivel ekvivalens**



**b)4123,1234, 1423, 1243 Tehát 4 db soros ütemezés van amivel ekvivalens**



**4. feladat (8 pont)**

Az alábbi ütemezésből egy művelet hiányzik a kérdőjelek helyén:

R1(C); W2(D); R2(B); ???; W1(A); R3(D); W2(C); W3(B)

Adjuk meg az összes lehetséges olyan olvasási műveletet, amelyet ha beillesztünk a hiányzó helyre, akkor az előálló ütemezés nem lesz konfliktus-sorbarendezhető. Indokoljuk is meg!

**A megoldást kérem ide írják:**

**R2(A) ->mivel R2(A)W1(A) nem felcserélhető**

**R3(C)->mivel R3(C)W2(C) nem felcserélhető**

**R3(A)->mivel R3(A)W1(A) nem felcserélhető**

**R1(D)->mivel W2(D)R1(D) nem felcserélhető**

**Összesen négy ilyen olvasást tudunk megadni!**

**Számítógépes feladatok**

**5. feladat (8 pont, ez a kötelezően megoldandó)**

Küldjék el egy külön szöveges állományban a zh2\_kotelezo\_feladat.txt állományban leírt feladat megoldását. A forráskódot és az alábbi futtatás eredményét kell elküldeni.

CALL print\_histogram('nikovits','test2','yr');

**Mellékelve beadando\_script.txt néven!**

A további lekérdezésekhez (6. és 7. feladat) az alábbi, NIKOVITS felhasználó tulajdonában levő táblákat kell használni, vagyis ne hozzanak létre saját táblákat.

CIKK(ckod, cnev, **szin**, suly)

SZALLITO(szkod, sznev, statusz, telephely)

PROJEKT(pkod, pnev, **helyszin**)

SZALLIT(szkod, ckod, pkod, mennyiseg, datum)

**6. feladat (4 + 4 pont)**

Adjuk meg a pécsi helyszínű **projektekhez** (**helyszin**=’Pecs’) szállított fekete színű cikkek összmennyiségét. (A SZALLIT tábla Mennyiseg oszlopát kell összegezni.)

**a)** Adjuk meg úgy a lekérdezést hintek segítségével, hogy a végrehajtási tervben minden join művelet SORT-MERGE join legyen, és indexet ne használjon a rendszer.

**b)** Adjuk meg úgy a lekérdezést hintek segítségével, hogy a végrehajtási tervben minden join művelet NESTED LOOP legyen, és a végrehajtás egy indexet használjon.

A **lekérdezést**, annak **végeredményét** és a **végrehajtási tervet** is el kell küldeni.

**A megoldást kérem ide másolják be:**

**a)**

---SCRIPT:

SELECT /\*+ USE\_MERGE(c,sz,p) NO\_INDEX(p) NO\_INDEX(sz) NO\_INDEX(c)\*/ sum(sz.mennyiseg)

FROM nikovits.projekt p, nikovits.szallit sz, nikovits.cikk c

WHERE p.helyszin = 'Pecs'

AND c.szin = 'fekete'

AND p.pkod = sz.pkod

AND c.ckod = sz.ckod;

---EREDMENY:

5457

---VEGREHELYTASI TERV

SELECT STATEMENT + +

SORT + AGGREGATE +

MERGE JOIN + +

SORT + JOIN +

MERGE JOIN + +

SORT + JOIN +

TABLE ACCESS + FULL + NIKOVITS.CIKK

SORT + JOIN +

TABLE ACCESS + FULL + NIKOVITS.SZALLIT

SORT + JOIN +

TABLE ACCESS + FULL + NIKOVITS.PROJEKT

**b)**

---SCRIPT:

SELECT /\*+ USE\_NL(c,sz,p) INDEX(p) NO\_INDEX(sz) NO\_INDEX(c)\*/ sum(sz.mennyiseg)

FROM nikovits.projekt p, nikovits.szallit sz, nikovits.cikk c

WHERE p.helyszin = 'Pecs'

AND c.szin = 'fekete'

AND p.pkod = sz.pkod

AND c.ckod = sz.ckod;

---EREDMENY:

5457

---VEGREHELYTASI TERV

SELECT STATEMENT + +

SORT + AGGREGATE +

NESTED LOOPS + +

NESTED LOOPS + +

NESTED LOOPS + +

TABLE ACCESS + FULL + NIKOVITS.CIKK

TABLE ACCESS + FULL + NIKOVITS.SZALLIT

INDEX + UNIQUE SCAN + NIKOVITS.P\_PKOD

TABLE ACCESS + BY INDEX ROWID + NIKOVITS.PROJEKT

**7. feladat (8 pont)**

Adjunk meg egy olyan lekérdezést (hintekkel együtt, ha szükséges), aminek az alábbi lesz a végrehajtási terve:

SELECT STATEMENT + +

SORT + GROUP BY +

CONCATENATION + +

TABLE ACCESS + BY INDEX ROWID BATCHED + NIKOVITS.SZALLIT

INDEX + RANGE SCAN + NIKOVITS.SZT\_PKOD

TABLE ACCESS + BY INDEX ROWID BATCHED + NIKOVITS.SZALLIT

INDEX + RANGE SCAN + NIKOVITS.SZT\_SZKOD

Adjuk meg a **lekérdezést**, és a **végrehajtási tervet** is, még akkor is, ha nem teljesen egyezik meg a végrehajtási terv az elvárttal. A feladatra részpontszám is szerezhető.

**A megoldást kérem ide másolják be:**

---SCRIPT:

select /\*+ index(sz) use\_concat use\_sort(sz) \*/ mennyiseg

from nikovits.szallit sz where pkod = 1 or szkod = 2

group by mennyiseg

order by mennyiseg;

---Lekérdezési terv:

SELECT STATEMENT + +

SORT + GROUP BY +

CONCATENATION + +

TABLE ACCESS + BY INDEX ROWID BATCHED + NIKOVITS.SZALLIT

INDEX + RANGE SCAN + NIKOVITS.SZT\_PKOD

TABLE ACCESS + BY INDEX ROWID BATCHED + NIKOVITS.SZALLIT

INDEX + RANGE SCAN + NIKOVITS.SZT\_SZKOD