

EXAM 2 2021

23.1.22.

(1)

(a) SELECT mem.name, mem.surname, mem.phone,
borrow.count FROM Members mem

JOIN (SELECT) FROM Borrow brw
WHERE date >= '1.1.2021' AND
brw.bid IN (SELECT book.bid FROM
Books book WHERE author LIKE '%Knuth')
Group By mid
HAVING COUNT(kid) > 5) borrow
ON mem.mid = borrow.mid

(b)

BOOKS

bid	author	title	publisher	year
-BID	LIKE "%Knuth"			

MEMBERS

mid	name	surname	street	city	phone	P.COUNT(BID)
-MID	P.G	P.G			P.G	

BORROW

mid	bid	mid	sid	date	
-RID	-BID	G.-MID		>= '1.1.2021'	

conditions

Count - RID > 5

(c)

(2)

	b ₀	b ₁	level = 0, H=4	
I(5):	000	00	[0 * 32]	ex-next → X2
S mod 4 = 1	001	01	[1 9 32 33 13] → [5 13]	
↓	010	10	[2 6 34 22]	
SPLIT	011	11	[3 7 11 15] → [23]	
	100	00	[12]	
	101	01	[21 37 5]	

$$0 \bmod 8 = 0; 12 \bmod 8 = 4; 32 \bmod 8 = 0$$

$$I(22): 22 \bmod 4 = 2$$

$$I(13): 13 \bmod 4 = 1$$

$$I(23): 23 \bmod 4 = 3 \Rightarrow \text{SPLIT}$$

$$1 \bmod 8 = 1; 9 \bmod 8 = 1; 21 \bmod 8 = 5; 37 \bmod 8 = 5$$

$$5 \bmod 8 = 5; 13 \bmod 8 = 1$$

- (b) - start from root and look for appropriate
leaf L to add the new index and add
- if L not full, done
- if it is, we break up in two ~ equal
halves L and L2
- copy up key
- recursively on index updates (move up when nec!)

(3)

(a) ii

Borrow.bid, Borrow.date

(o)

date > 2020/1/1

(borrow)

mid

o (Members))

Borrow.mid = Members.mid "city_topper"

(b)

bid, date



mid mid

o

date > 2020/1/1

o

city_topper

Borrow

Members

unclustered

Scan Borrow T1

Members as data
cause hash index
on mid

• Scan Borrow with B+ index over date:

- assuming unclustered

$$(3 + 3000000 * \frac{1}{22}) = \\ = 136367 \text{ pgs reads}$$

- assuming range

$$\cdot \text{Write T2: } 136367 / 200 = 682 \text{ pg writes} \\ (\text{rec per pg})$$

of years [2020-2021]

$$\cdot \text{Scan Members: } 12500 \text{ pg reads}$$

- assuming 50 cities

$$\cdot \text{write T1: } 12500 * \frac{1}{50} = 250 \text{ pgs writes}$$

assuming 50 cities

$$T1: 50 * 100 = 2000 \text{ records}$$

hashing on key

• Index nested loops join or hash index

on Members.mid: for both

$$M + (\# \text{ of pg of outer}) * (1.2 + \text{cost of matching tuples})$$

* # of rec per pg

$$682 + 682 * 200 * (1.2 + \dots) = 300762 \text{ reads}$$



we need 2000 records? $\text{clust} = 50 \text{ pg, unclus} = 2000 \text{ pg}$

total reads:

total cost: 440 361

(d) Sort By ALTERNATIVES: (2 to 4) sorting and
hashing

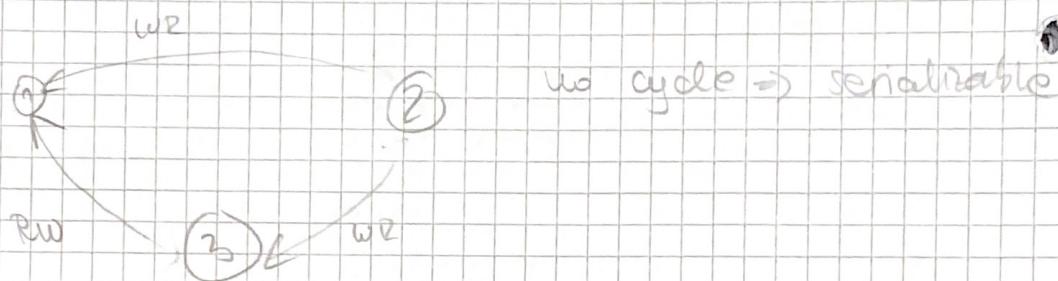
(e)

(a) RW: $R_3(x), W_1(x)$

WR: $(W_2(x), R_3(x)) (W_2(y), R_3(y)), (W_2(x), R_1(x))$

WW: $(W_2(x), W_3(x))$

(b)



conflict equivalent to $T_2 \rightarrow T_3 \rightarrow T_1$