National University of Computer and Emerging Sciences
Lahore Campus

Advance Database Concepts (CS4064)

Date: Fri, 31 May 2024

Course Instructor(s)

Muhammad Ishaq Raza

Muhammad Naveed

Final Exam

Total Time (Hrs.): 3

Total Marks: 60

Total Questions: 7

_____________ _____________ _________________________

Roll No Section Student Signature

Do not write below this line.

Note: Please ensure that you attempt all questions and their respective parts in the given order.

SOLUTION

CLO # 3: To develop a solution for given scenario/challenging problem in the domain of DB systems.

Q. No 1: Consider the following part of library database schema and the query in *SQL* and *RA*:

Book (BookID, Title, Category, Publisher, PublishYear)
Author (AuthorID, AuthorName, Gender, Email, OriginCity)
BookAuthor (BookID, AuthorID)

SELECT Title, AuthorName, Publisher
FROM Author A JOIN BookAuthor BA ON A.AuthorID=BA.AuthorID JOIN Book B ON B.BookID=BA.BookID
WHERE Gender= 'Male' AND Category= 'Education';

$\pi_{\text{Title, AuthorName, Publisher}} (\sigma_{\text{Gender='Male'} \wedge \text{Category='Education'}} (\text{Author} * \text{BookAuthor} * \text{Book}))$

Your task is to optimize this query and draw the best possible query tree for this query. Take appropriate database statistics to support your answer. [10]

Ans:

CLO # 3: To develop a solution for given scenario/challenging problem in the domain of DB systems.

Q. No 2: [10]

a. Consider the above library database schema and the query in *SQL/RA* given in *Q#1*. Assume that the frequency of access of this query is very high. Which attributes are more appropriate to create indexes for an efficient execution strategy to improve the performance of this query?

b. Consider the above library database schema and assume that *Book, Author* and *BookAuthor* tables have *100000*, *100000* and *50000* rows respectively. Estimate the potential *Join Cardinality* (*jc*) of *Book* $\bowtie_{BookID=BookID}$ *BookAuthor* (i.e., max number of rows resulting from the inner join of these two tables). Justify your answer.

Ans:

a) Book.Category, Author.Gender, and joining columns of all 3 tables.

b) Joining will yield a table of at most 50000 tuples, since BookID is a key for Book table.

js= 1/MAX(NDV(bookID, Book), NDV(bookID, BookAuthor))= 1/100000 = 0.00001

jc= js * |Book| * |BookAuthor|= 0.00001 * 100000 * 50000 = 50,000

CLO # 3: To develop a solution for given scenario/challenging problem in the domain of DB systems.

Q. No 3: Assume: A block size is *B= 1024* bytes, file has *r= 1,000,000* records, each record is *100* bytes long, a block pointer is *P= 10* bytes, a record pointer is P_R= *11* bytes, and a key field for the index is 6 bytes long. A database system uses a B+-trees index on key field. A leaf node and non-leaf node are one block in size and contain as many keys (and appropriate pointers) as will fit in a block. How many blocks will this index use? Also estimate the number of block accesses needed to search for and retrieve a record from the file given its key value using the B+-tree index. Show your working. [10]

Ans:

order of p: (p * 10) + (p-1) * 6) < 1024, which gives us order p = (1024 + 6)/16 = 64. order of p_{leaf}: (p * (6+11) + 10) < 1024, which gives us order p_{leaf} = (1024 - 10)/17 = 59.

This means the leaves (b1) will require ceiling (1,000,000/59) = 16950 blocks.

Thus, our second level (b2) above the leaves will require ceiling (16950/64) = 265 blocks.

The third level (b3) above that will require ceiling (265/64) = 5 blocks.

The fourth level (b4) above that will require 1 block.

The total blocks this index uses are 17,221 blocks.

Block access cost to search a key value= x+1= 4+1= 5

CLO # 2: Apply the models and approaches to become enabled to select and apply appropriate methods for a particular case.

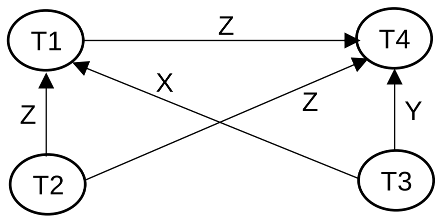
Q. No 4: Consider the following schedule: [5]

S: r1(X), r2(Z), w1(Z), r3(X), r3(Y), w1(X), w3(Y), r4(Y), w4(Z), w4(Y).

Draw the serializability (precedence) graph for this schedule. State whether this schedule is conflict-serializable (correct) or not. If the schedule is conflict-serializable, write down the equivalent serial schedule(s) otherwise explain why it is not. *Also state whether this schedule is view-serializable or not.*

Ans:

Ans: It is conflict-serializable and equivalent serial schedules are T2→T3→T1→T4 and T3→T2→T1→T4. It is also view-serializable.



CLO # 2: Apply the models and approaches to become enabled to select and apply appropriate methods for a particular case.

Q. No 5: Consider the following schedule of actions: [10]

S: r1(X), r2(Z), w1(Z), r3(X), r3(Y), w1(X), w3(Y), r2(Y), c3, c2, c1, w4(Z), w4(Y), c4.

For each of the following concurrency control mechanisms, describe how the concurrency control mechanism handles the schedule. Assume that the timestamp of transaction *Ti* is *i*. For lock-based concurrency control mechanisms, add lock and unlock requests to the above schedule of actions as per the locking protocol. The DBMS processes actions in the order shown. If a transaction is blocked, assume that all its actions are queued until it is resumed; the DBMS continues with the next action (according to the listed schedule) of an unblocked transaction.

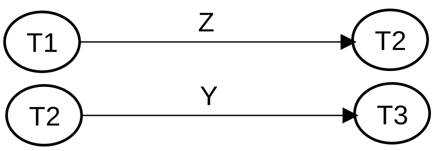
a. Rigorous 2PL with timestamps used for deadlock detection (Use wait-for-graph to deal with deadlock)

b. Basic Timestamp Ordering (Assume T1 < T2 < T3)

Ans: a)

T_1	**T_2**	**T_3**	**T_4**
s1-lock(X) r1(X)			
	s2-lock(Z) r2(Z)		
x1-lock(Z) ...**wait** for T2			
		s3-lock(X) r3(X) s3-lock(Y) r3(Y) x3-lock(Y) ...**upgrade-lock** w3(Y)	
	S2-lock(Y) ...**wait** for T3		
		c3, release all locks. unlock3(X) unlock3(Y)	
	r2(Y) ...**wake-up** **c2**, release all locks. unlock2(Z) unlock2(Y)		
w1(Z) ...**wake-up** x1-lock(X) ...**upgrade-lock** w1(X) **c1**, release all locks. unlock1(Z) unlock1(X)			
			x4-lock(Z) w4(Z) x4-lock(Y) w4(Y) **c4**, release locks. unlock4(Z) unlock4(Y)

Wait-for-graph:



b)

T1	**T2**	**T3**	**T4**	**X**		**Y**		**Z**	
				RTS	*WTS*	*RTS*	*WTS*	*RTS*	*WTS*
				T0	{ }	**T0**	{ }	**T0**	{ }
r1(X)				{T1}					
	r2(Z)							{T2}	
w1(Z)									
abort T1 as									
RTS(Z)>TS(T1)				{T3}					
		r3(X)				{T3}			
		r3(Y)					T3		
		w3(Y)							
	r2(Y)								
	abort T2 as								
	WTS(Y)>TS(T2)								
		c3							
			w4(Z)						T4
			w4(Y)				T4		
			c4						
restart T1									
with new TS.									
	Restart T2								
	with new TS.								

CLO # 2: Apply the models and approaches to become enabled to select and apply appropriate methods for a particular case.

Q. No 6: Consider the following log at the point of system crash. Suppose that we use ARIES recovery algorithm to answer the following questions. [5]

LSN	Last_LSN	Trans_ID	Type	Page_ID	Other_Info
1	0	T1	Update	A	...
2	0	T2	Update	C	...
3	1	T1	Commit		...
4	2	T2	Update	A	...
5	begin checkpoint				
6	end checkpoint				
7	4	T2	Commit		...
8	0	T3	Update	B	...
9	0	T4	Update	C	...
10	8	T3	Update	A	...
11	9	T4	Commit		...

a. Show the contents of transaction and dirty page table at the time of checkpoint. [1]

b. What is done during Analysis? Be precise about the points at which Analysis begins and ends and show the contents of transaction and dirty page table reconstructed in this phase. [2]

c. What is done during Redo? Be precise about the points at which Redo begins and ends. [1]

d. What is done during Undo? Be precise about the points at which Undo begins and ends. [1]

Ans: a)

Trans_ID	LSN	Status
T1	**3**	**commit**
T2	**4**	**in-progress**

Page_ID	LSN
A	**1**
C	**2**

b) Analysis: Start from 5 *(i.e. begin checkpoint)* till LSN# 11

Transection Table

Trans_ID	LSN	Status
T1	**3**	**commit**
T2	**7**	**commit**
T3	**10**	**in-progress**
T4	**11**	**commit**

Dirty Page Table

Page_ID	LSN
A	**1**
B	**8**
C	**2**

c) Begin_LSN (1) End_LSN(10)
LSNs (1,2,4,8,9,10) update the corresponding pages (A, C, A, B, C, A)

c) Begin_LSN (10) End_LSN (8)
LSNs (10,8) corresponding pages (A, B)
Undo will perform on T3 transaction.

CLO # 1: Understanding advance data models, technologies, and approaches for building DB systems.

Q. No 7: [10]

a. Discuss with example, the concept of data fragmentation and data allocation in distributed database design. Why is fragmentation a useful concept in distributed database design?

b. Discuss the partition tolerance and eventual consistency properties of CAP theorem.

Ans: