



United International University (UIU)

Dept. of Computer Science & Engineering (CSE)

Final Exam, Trimester: Summer 2022

Course Code: CSE-3521

Course Title: Database Management Systems

Total Marks: 40

Duration: 2 hours

Any examinee found adopting unfair means will be expelled from the trimester / program as per UIU disciplinary rules.

1.	<p>a. Write briefly about sparse and dense index.</p> <p>b. Construct a B+ tree from the following keys. Assume that the tree is initially empty and values are added sequentially one by one.</p> <p>(Order 4) 5, 50, 100, 25, 40, 45, 150, 80, 30, 15, 35</p>	3+7																																	
2.	<p>Consider an extendible hashing scheme where the bucket capacity is 3 and the initial local and global depth are 1. Insert the following records in the hash table showing all the states for each insertion. Assume that the LSB (least-significant bit) is being checked to find the directory for a record.</p> <table><tr><th>Pointer</th><th>Key_value</th><th>Hash(key_value)</th></tr><tr><td>Pointer 1</td><td>7856</td><td>13</td></tr><tr><td>Pointer 2</td><td>4256</td><td>2</td></tr><tr><td>Pointer 3</td><td>8954</td><td>18</td></tr><tr><td>Pointer 4</td><td>4523</td><td>25</td></tr><tr><td>Pointer 5</td><td>1593</td><td>8</td></tr><tr><td>Pointer 6</td><td>7524</td><td>15</td></tr><tr><td>Pointer 7</td><td>2459</td><td>10</td></tr><tr><td>Pointer 8</td><td>5648</td><td>5</td></tr><tr><td>Pointer 8</td><td>9548</td><td>21</td></tr><tr><td>Pointer 10</td><td>3694</td><td>1</td></tr></table>	Pointer	Key_value	Hash(key_value)	Pointer 1	7856	13	Pointer 2	4256	2	Pointer 3	8954	18	Pointer 4	4523	25	Pointer 5	1593	8	Pointer 6	7524	15	Pointer 7	2459	10	Pointer 8	5648	5	Pointer 8	9548	21	Pointer 10	3694	1	10
Pointer	Key_value	Hash(key_value)																																	
Pointer 1	7856	13																																	
Pointer 2	4256	2																																	
Pointer 3	8954	18																																	
Pointer 4	4523	25																																	
Pointer 5	1593	8																																	
Pointer 6	7524	15																																	
Pointer 7	2459	10																																	
Pointer 8	5648	5																																	
Pointer 8	9548	21																																	
Pointer 10	3694	1																																	
3.	<p>a) Consider the following relation R1 and set of functional dependencies F1</p> <p>R= { A, B, C, D, E, I }</p> <p>F= { $A \rightarrow C$, $AB \rightarrow C$, $C \rightarrow DI$, $CD \rightarrow I$, $EC \rightarrow AB$, $EI \rightarrow C$ }</p> <p>i) Determine all the candidate keys for the relation R.</p> <p>ii) Find the attribute closure for (ACD) and (BCI) for the relation R.</p> <p>iii) Find the maximum normalized form (NF) of relation R</p>	3+ 1+ 2																																	

	<p>b) Consider the following relation R2 and set of functional dependencies F2 R={ A, B, C, D, E, F, G, H, I, J } F={ AB→C, AD→GH, BD→EF, A→I, H→J, I→BD }</p> <p>i) Check whether A→C and I→G are valid functional dependencies or not.</p> <p>ii) Check if G is a prime attribute or not.</p>	2+2																																																																	
4.	<p>a) Write down when a schedule will be considered as view serializable with proper examples. Mention the problems of concurrent schedule handling.</p> <p>b) Draw the precedence graph and show the following schedule is conflict serializable or not. If it is conflict serializable, find out the corresponding serial schedule.</p> <table><tr><td>T1</td><td>T2</td><td>T3</td><td>T4</td><td>T5</td></tr><tr><td></td><td></td><td>read(Q)</td><td></td><td></td></tr><tr><td>read(R)</td><td></td><td></td><td></td><td></td></tr><tr><td>write(S)</td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td>read(S)</td><td></td></tr><tr><td></td><td>write(P)</td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td>write(R)</td></tr><tr><td></td><td></td><td>read(P)</td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td>read(Q)</td></tr><tr><td></td><td></td><td></td><td></td><td>read(L)</td></tr><tr><td></td><td></td><td></td><td>read(T)</td><td></td></tr><tr><td></td><td>read(R)</td><td></td><td></td><td></td></tr><tr><td></td><td></td><td>write(T)</td><td></td><td></td></tr></table>	T1	T2	T3	T4	T5			read(Q)			read(R)					write(S)								read(S)			write(P)								write(R)			read(P)							read(Q)					read(L)				read(T)			read(R)						write(T)			3+7
T1	T2	T3	T4	T5																																																															
		read(Q)																																																																	
read(R)																																																																			
write(S)																																																																			
			read(S)																																																																
	write(P)																																																																		
				write(R)																																																															
		read(P)																																																																	
				read(Q)																																																															
				read(L)																																																															
			read(T)																																																																
	read(R)																																																																		
		write(T)																																																																	