SPRING END SEMESTER EXAMINATION-2023
4th Semester, B.Tech (Programme)

SUBJECT : DATABASE MANAGEMENT SYSTEM

CODE: CS 2004/ CS-2004

(For 2021 Admitted Batches)

Time: 3 Hours **Full Marks: 50**

Answer any SIX questions.
Question paper consists of four SECTIONS i.e. A, B, C and D.
Section A is compulsory.
Attempt minimum one question each from Sections B, C, D.
The figures in the margin indicate full marks.
Candidates are required to give their answers in their own words as far as practicable and all parts of a question should be answered at one place only.

SECTION-A (Learning levels 1 and 2)

1.		Answer the following questions.	[1 × 10]
	(a)	Differentiate between Cartesian product and Join operation with suitable example**.**	1
	Ans	On applying CARTESIAN PRODUCT(✕) on two relations that is on two sets of tuples, it will take every tuple one by one from the left set(relation) and will pair it up with all the tuples in the right set(relation). So, the Cartesian product of two relation A(R1, R2, R3, …, Rp) with degree p, and B(S1, S2, S3, …, Sn) with degree n, is a relation C(R1, R2, R3, …, Rp, S1, S2, S3, …, Sn) with degree p + n attributes. Cartesian Product is a binary set operation means, at a time we can apply the operation on two relations. But the two relations on which we are performing the operations do not have the same type of tuples, which means Union compatibility (or Type compatibility) of the two relations is not necessary. Consider two relations STUDENT(SNO, FNAME, LNAME) and DETAIL(ROLLNO, AGE) below:	

SNO	FNAME	LNAME
1	Albert	Singh
2	Nora	Fatehi

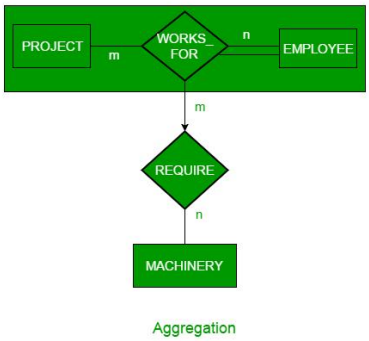
ROLLNO	AGE
5	18
9	21

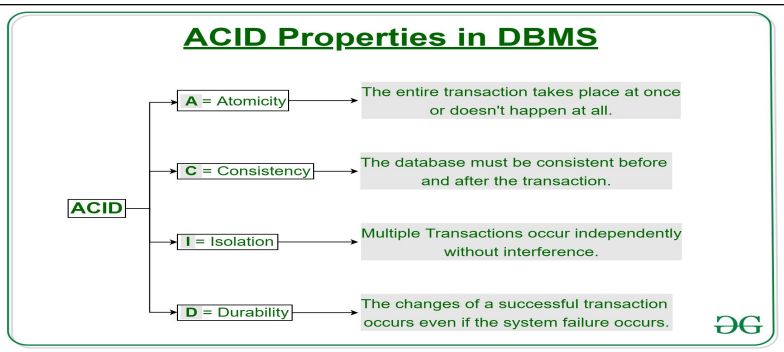
On applying CROSS PRODUCT on STUDENT and DETAIL:

STUDENT ✕ DETAILS

SNO	FNAME	LNAME	ROLL NO	AGE
1	Albert	Singh	5	18
1	Albert	Singh	9	21
2	Nora	Fatehi	5	18
2	Nora	Fatehi	9	21

Join operation(Inner and Outer join) is essentially a cartesian product followed by a selection criterion.Join operation denoted by *Select Operation*

		JOIN operation also allows joining variously related tuples from different relations. In an inner join, only those tuples that satisfy the matching criteria are included, while the rest are excluded.In an outer join, along with tuples that satisfy the matching criteria, we also include some or all tuples that do not match the criteria.	
	(b)	What do you mean by inconsistent database? Discuss with suitable example.	.5 +.5
	Ans	Consider the following transaction T consisting of T1 and T2: Transfer of 100 from account X to account Y. **Before:** X : 500 | Y: 200 Transaction T **T1** | **T2** Read (X) | Read (Y) X: = X − 100 | Y: = Y + 100 Write (X) | Write (Y) **After:** X : 400 | Y : 300 If the transaction fails after completion of T1 but before completion of T2.(say, after write(X) but before write(Y)), then the amount has been deducted from X but not added to Y. This results in an inconsistent database state. Therefore, the transaction must be executed in its entirety in order to ensure the correctness of the database state.	
	(c)	What is aggregation? How is it represented using ER diagram? Give example to support your answer**.**	.5 +.5
	Ans	An ER diagram is not capable of representing relationship between an entity and a relationship which may be required in some scenarios. In those cases, a relationship with its corresponding entities is aggregated into a higher level entity. *Aggregation is an abstraction* through which we can represent relationships as higher level entity sets. For Example, Employee working for a project may require some machinery. So, REQUIRE relationship is needed between relationship WORKS_FOR and entity MACHINERY. Using aggregation, WORKS_FOR relationship with its entities EMPLOYEE and PROJECT is aggregated into single entity and relationship REQUIRE is created between aggregated entity and MACHINERY.  Aggregation	
	(d)	Given the relations Students(Name, Marks, SchoolNo) Schools (SchoolNo, SchoolName, Address) Which of the following queries cannot be expressed using the basic relational algebra operations (σ, π, ×, ∪, −)? (A) School Address of every student (B) Students whose name is same as their school name	1

		(C) The sum of all students' marks (D) Name of students of a given school	
	Ans	(C) The sum of all students' marks	
	(e)	**Consider the following table named Student in a relational database. The primary key of this table is rollNum.** rollNum | name | gender | marks 1 | Naman | M | 62 2 | Aliya | F | 70 3 | Aliya | F | 80 4 | James | M | 82 5 | Swati | F | 65 The SQL query below is executed on this database. 1. SELECT * 2. FROM Student 3. WHERE gender = 'M' AND 4. marks > 65; The number of rows returned by the query is ---------.	1
	Ans	Roll | Name | Gender | Marks 4 | James | M | 82 So the output of the given query is 1. Note: The query will return only a single row as both the select criteria are satisified for 4th row where rollNum is 4	
	(f)	Write the ACID properties of transaction with citing one example of 'I' property.	.5 +.5
	Ans	 This property ensures that multiple transactions can occur concurrently without leading to the inconsistency of the database state. Transactions occur independently without interference. Changes occurring in a particular transaction will not be visible to any other transaction until that particular change in that transaction is written to memory or has been committed. This property ensures that the execution of transactions concurrently will result in a state that is equivalent to a state achieved these were executed serially in some order. Let X= 500, Y = 500. Consider two transactions T and T''. T | T'' Read (X) | Read (X) X: = X*100 | Read (Y) Write (X) | Z: = X + Y Read (Y) | Write (Z) Y: = Y – 50 Write (Y) Suppose T has been executed till Read (Y) and then T'' starts. As a result,interleaving of operations takes place due to which T'' reads the correct value of X but the incorrect value of Y and sum computed by	

		T'': (X+Y = 50, 000+500=50, 500) is thus not consistent with the sum at end of the transaction: T: (X+Y = 50, 000 + 450 = 50, 450). This results in database inconsistency, due to a loss of 50 units. Hence, transactions must take place in isolation and changes should be visible only after they have been made to the main memory.	
	(g)	Select the TRUE statement from the following: i. A relation R is in 3NF if every non-prime attributes of R is completely functionally dependent on every key of R ii. Every relation in 3NF is also in BCNF iii. No relation can be in both 3NF and BCNF iv. Every relation in BCNF is also in 3NF	1
	Ans	(iv) is true, since BCNF is stricter than 3NF	
	(h)	**Given the following relation instance.** **X** | **Y** | **Z** 1 | 4 | 2 1 | 5 | 3 1 | 6 | 3 3 | 2 | 2 **Which of the following functional dependencies are satisfied by the instance?** (a) XY -> Z and Z -> Y (b) YZ -> X and Y -> Z (c) YZ -> X and X -> Z (d) XZ -> Y and Y -> X	1
	Ans	(b) YZ -> X and Y -> Z In the above question, Y uniquely determines X and Z, for a given value of Y you can easily find out values of X and Z. So, Y -> X and Y -> Z hold for above schema. From rule of augmentation we can say YZ->X	
	(i)	In a relation scheme R = (A, B, C, D, E, H) on which the following functional dependencies hold: {A–>B, BC–>D, E–>C, D–>A}, the candidate keys of R are_____ i. AE, BE ii. AE, BE, DE iii. AEH, BEH, BCH iv. AEH, BEH, DEH	1
	Ans	iv. AEH, BEH, DEH A set of attributes S is candidate key of relation R if the closure of S is all attributes of R and there is no subset of S whose closure is all attributes of R. Closure of AEH, i.e. AEH+ = {ABCDEH} Closure of BEH, i.e. BEH+ = {ABCDEH} Closure of DEH, i.e. DEH+ = {ABCDEH}	
	(j)	How many minimum internal nodes are required for 1000 leaf nodes of B+ tree with order P:8. (Assume Order P: maximum pointer per node.)	1
	Ans	Leaf level i.e. Level 4 has 1000, Level 3 has (1000/8)= 125 nodes, Level 2 has ceiling(125/8) = 16 nodes, Level 1 has (16/8)=2 nodes and level 0 has ceiling(2/8) = 1 node	
SECTION-B (Learning levels 1,2, and 3)			
2.	(a)	Why concurrency control is needed? Explain lost update, dirty read and incorrect summary problem with suitable example.	2+2

Ans

Concurrency control is an essential aspect of database management systems (DBMS) that ensures transactions can execute concurrently without interfering with each other. In a multi-user system, multiple users can access and use the same database at one time, which is known as the concurrent execution of the database. It means that the same database is executed simultaneously on a multi-user system by different users.

Concurrency Control is the working concept that is required for controlling and managing the concurrent execution of database operations and thus avoiding the inconsistencies in the database. Thus, for maintaining the concurrency of the database, we have the concurrency control protocols. The concurrency control protocols ensure the atomicity, consistency, isolation, durability and serializability of the concurrent execution of the database transactions. Therefore, these protocols are categorized as: (i) Lock Based Concurrency Control Protocol, (ii) Time Stamp Concurrency Control Protocol, and (iii) Validation Based Concurrency Control Protocol

When multiple transactions execute concurrently in an uncontrolled or unrestricted manner, then it might lead to several problems.

- Temporary Update Problem or Dirty Read Problem
- Incorrect Summary Problem
- Lost Update Problem
- Unrepeatable Read Problem
- Phantom Read Problem

Temporary update or dirty read problem occurs when one transaction updates an item and fails. But the updated item is used by another transaction before the item is changed or reverted back to its last value.

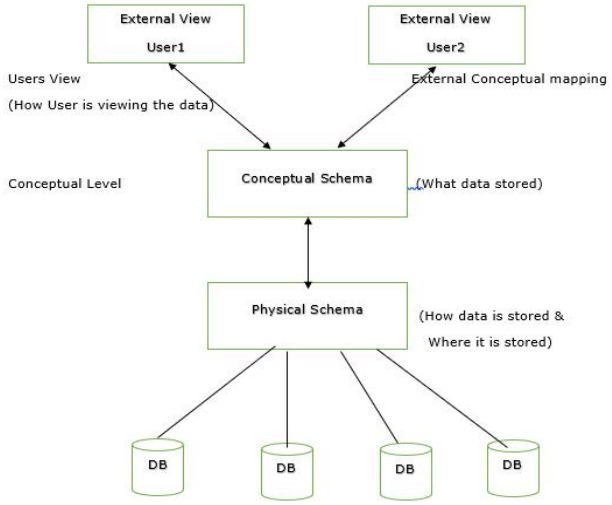
T1	T2
Read(X) X=X++ Write(X)	
	Read(X) X=X++ Write(X)
Read(Y)	

In the above example, if transaction 1 fails for some reason then X will revert back to its previous value. But transaction 2 has already read the incorrect value of X.

Incorrect Summary Problem: Consider a situation, where one transaction is applying the aggregate function on some records while another transaction is updating these records. The aggregate function may calculate some values before the values have been updated and others after they are updated.

T1	T2

		<table><tr><td>Read(X) X=X-N Write(X) Read(Y) Y=Y+N Write(Y)</td><td>Sum=0 Read(A) Sum=Sum+A Read(X) Sum=Sum+X Read(Y) Sum=Sum+Y</td></tr></table> In the above example, transaction 2 is calculating the sum of some records while transaction 1 is updating them. Therefore the aggregate function may calculate some values before they have been updated and others after they have been updated.	
	(b)	Describe the three-schema architecture. Why do we need mappings between schema levels? Also, differentiate between logical data independence and physical data independence	2+1+1
	Ans	The three schema architecture is also called ANSI/SPARC architecture or three-level architecture.This framework is used to describe the structure of a specific database system. The three schema architecture is also used to separate the user applications and physical database.The three schema architecture contains three-levels. It breaks the database down into three different categories: *(i) External level, (ii) Conceptual level, (iii) Internal level.* **External/ View level:** This is the highest level of database abstraction. It includes a number of external schemas or user views. This level provides different views of the same database for a specific user or a group of users. An external view provides a powerful and flexible security mechanism by hiding the parts of the database from a particular user. **Conceptual or Logical level:** This level describes the structure of the whole database. It acts as a middle layer between the physical storage and user view. This level describes the structure of the whole database. It acts as a middle layer between the physical storage and user view. It explains what data to be stored in the database, what the data types are, and what relationship exists among those data. There is only one conceptual schema per database.	



Internal or Physical level: This is the lowest level of database abstraction. It describes how the data is stored in the database and provides the methods to access data from the database. It allows viewing the physical representation of the database on the computer system**.**

The interface between the conceptual and internal schema identifies how an element in the conceptual schema is stored and how it may be accessed. The internal schema not only defines different stored record types, but also specifies what indices exist, how stored fields are represented.

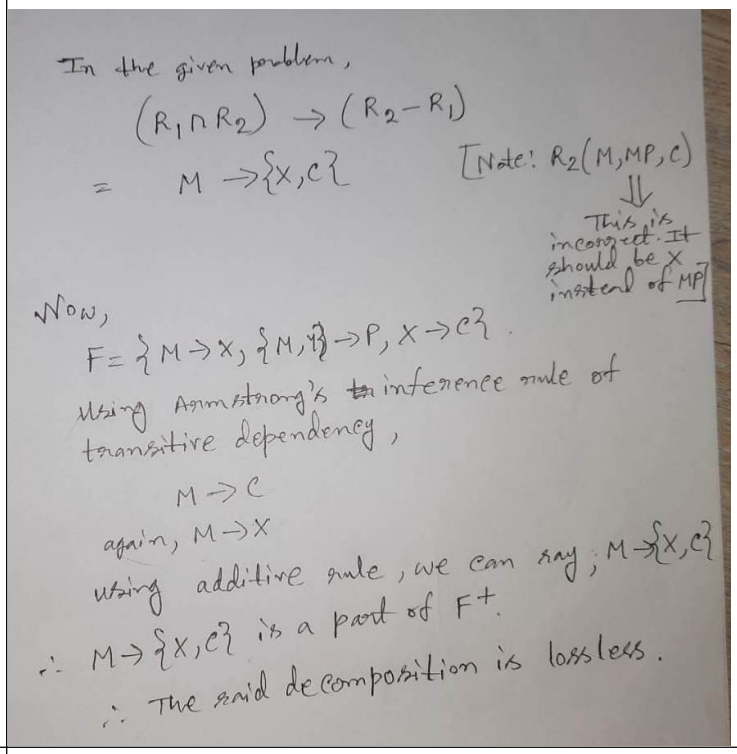
Why do we need mappings between schema levels?

Mapping between schema levels are needed to ensure that data is organized and structure in consistent and logical way. Schema levels refers to different level of abstraction in a database schema such as the conceptual, logical, and physicals levels. There are basically two types of mapping in the database architecture: (i) Conceptual/ Internal Mapping, and (ii) External / Conceptual Mapping

Physical Data Independence	**Logical Data Independence**
It mainly concern about how the data is stored into the system.	It mainly concerned about the structure or the changing data definition.
Any change at the physical level, does not require to change at the application level.	The change in the logical level requires a change at the application level.
The modifications made at the internal level may or may not be needed to improve the performance of the structure.	The modifications made at the logical level is significant whenever the logical structure of the database is to be changed.
Example: Change in compression techniques, Hashing algorithms and storage devices etc.	Example: Add/Modify or Delete a new attribute.

3.	(a)	Consider the relation E = (P, Q, R, S, T, U) having set of Functional	1+1+1+1

		Dependencies (FD). **P → Q, P → R ,QR → S , Q → T ,QR → U , PR → U** Infer the following members of Axioms applying Armstrong's Axioms and Additional inference rule. 1. **P → T** **2. PR → S** **3. QR → SU** **4. PR → SU**	
	Ans	**Solution:** **1. P → T** In the above FD set, P → Q and Q → T **So, Using Transitive Rule: If {A → B} and {B → C}, then {A → C}** ∴ If P → Q and Q → T, then **P → T.** **P → T** **2. PR → S** In the above FD set, P → Q As, QR → S **So, Using Pseudo Transitivity Rule: If{A → B} and {BC → D}, then {AC → D}** ∴ If P → Q and QR → S, **then PR → S.** **PR → S** **3. QR → SU** In above FD set, QR → S and QR → U **So, Using Union Rule: If{A → B} and {A → C}, then {A → BC}** ∴ If QR → S and QR → U, **then QR → SU.** **QR → SU** **4. PR → SU** **So, Using Pseudo Transitivity Rule: If{A → B} and {BC → D}, then {AC → D}** ∴ If PR → S and PR → U, **then PR → SU.** **PR → SU**	
	(b)	Given the following schemas, give the relational algebra and domain relational calculus expression for the queries. employee (person_name, street, city) works (person_name, company_name, salary) company (company_name, city) (i) Find the names of all employees whose salary is greater than 100000. (ii) Find the names of all employees who is from Mumbai and works in Delhi.	2+2
	Ans	**Solution:** **Relational Algebra Expressions:** (i) $\Pi_{\text{person-name}}(\sigma_{\text{salary} > 100000}(\text{employee} \bowtie_{\text{employee.person\_name} = \text{works.person\_name}} \text{works}))$ (ii) $\Pi_{\text{person-name}}(\sigma_{\text{employee.city} = \text{'Mumbai'} \wedge \text{company.city} = \text{'Delhi'}}(\text{employee} \bowtie_{\text{employee.person\_name} = \text{works.person\_name}} \text{works} \bowtie_{\text{works.company\_name} = \text{company.company\_name}} \text{company}))$ **Domain Relational Calculus Expressions:** (i) {x | (∃x) (∃m) (∃p) (employee(xyz) AND works(mnp) AND p > 100000 AND x = m)} (ii) {x | (∃x) (∃z) (∃m) (∃n) (∃q) (∃r) (employee(xyz) AND works(mnp) AND company(qr) AND z = 'Mumbai' AND r = 'Delhi' AND x = m AND n = q)}	
		SECTION-C (Learning Levels 3 and 4)	
4.	(a)	Consider the following relation R(M, Y, P, X, C) and with the following dependencies : F = { M→X, {M,Y} → P, X→C} Consider the decomposition D = { R1 (M, Y, P), R2 (M, MP, C) }. Check weather this decomposition is lossless or losy	2+2

		decomposition in step by step manner.	
	Ans	A decomposition $D = \{R_1, R_2\}$ of R has the lossless (nonadditive) join property with respect to a set of functional dependencies F on R *if and only if* either • The FD $((R_1 \cap R_2) \rightarrow (R_1 - R_2))$ is in F^+, or • The FD $((R_1 \cap R_2) \rightarrow (R_2 - R_1))$ is in F^+  	
	(b)	What is 4NF? Give an example of a relation schema R and a set of dependencies such that R is in BCNF, but not in 4NF.	1+3
	Ans	**Definition.** A relation schema R is in **4NF** with (that includes functional dependencies and multiv *nontrivial* multivalued dependency $X \twoheadrightarrow Y$ in F^+, X is **EMP** ENAME | PNAME | DNAME Smith X John Smith Y Anna Smith X Anna Smith Y John The EMP schema is in BCNF because no functional dependencies hold in EMP. the MVDs ENAME $\twoheadrightarrow$ PNAME and ENAME $\twoheadrightarrow$ DNAME (or ENAME $\twoheadrightarrow$ PNAME | DNAME) hold in the EMP relation. But, it is not in 4NF because in the nontrivial MVDs ENAME is not a superkey of EMP.	
5.	(a)	Find the canonical cover of the following FDs on R(VWXYZ) FD: V→W, VW→X, Y→VWX	[4]
	Ans	Step 1 -> Separate all right hand side value so that it become atomic. Step -1 Step- 2 Step-3	

V->W
V->WV
W->X
VW->X
Y->V
Y->V

1. $V^+=\{VWX\}$ $V^+=\{V\}$

2. $VW^+=\{VWX\}$ $VW^+=\{VW\}$

Y->W ✗

Y->X ✗

3. $Y^+=\{YVWX\}$ $Y^+=\{YWX\}$

4. $Y^+=\{YVWX\}$ $Y^+=\{YVXW\}$

5. $Y^+=\{YVWX\}$ $Y^+=\{YVWX\}$

Step-2 -> Check for each of the value LHS using closure from FD's (one time including it other time ignoring it).
If match means it is redundant.

Step-4 In step 4, we will check for redundancy in left hand side

V->W
V->X
Y->V

Of step 3

$VW^+=\{VWX\}$
$V^+=\{VWX\}$

Final(Combine)

V -> WX
Y-> V

We can get the desire value of VW from VW from V only then there is no need of w , so remove it . .

(b) Which normal form is considered adequate for normal database design ? Consider a relation with set of functional dependencies (FD) as R(ABCDEF).
FD= AB->CDEF, C->A, D->B, C->D, E->F and B->E.
What is the normal form of the relation ? 1+3

3NF is sufficient because because most of the 3NF tables are free of insertion, update, and deletion anomalies. Moreover, 3NF always ensures functional dependency preserving and lossless.

R(ABCDEF)

ıdidate keys = AB, AD,C
ime attributes will be { A,B,C,D} and Non-Prime will

FD	AB -> CDEF	C -> A	D -> B	C -> D	E -> F	**B -> E**
BCNF	Yes	Yes	No	Yes	No	No
3NF	Yes	Yes	Yes	Yes	No	No
2NF	Yes	yes	yes	yes	yes	No(PD)

So we can say that the highest normal form of above FD's is 1NF only.

6. (a) Consider the following two transactions : [4]

```
T 1 :   read ( A ) ;
        read ( B ) ;
        if A = 0, then
        B : = B + 1 ;
        write ( B )
T 2 :   read ( B ) ;
        read ( A ) ;
```

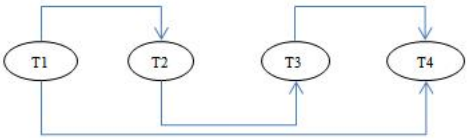
		if B = 0, then A : = A +1 ; write (A) Add lock and unlock instructions to transactions T 1 and T 2 , so that they observe the two-phase locking protocol. Can the execution of these transactions result in a deadlock ?	

	Ans	**Evaluation Scheme:** Full mark for the correct answer. Stepwise mark may be awarded based on the partial correctness of the solution.
		Solution: The lock and unlock instructions to transaction T1 and T2 is depicted below with the consideration of two-phase locking protocol.
		<table><tr><th>T1</th><th>T2</th></tr><tr><td>Lock-S(A)
		Read(A)
		Lock-X(A)
		Read(B)
		If (A = 0) THEN
		B := B +1
		Write(B)
		Unlock(A)
		Unlock(B)</td><td></td></tr><tr><td></td><td>Lock-S(B)
		Read(B)
		Lock-X(A)
		Read(A)
		If (B = 0) THEN
		A := A +1
		Write(A)
		Unlock(B)
		Unlock(A)</td></tr></table>The above schedule does not result into deadlock as none of the transactions are waiting for the shared resources to unlock the lock.	

	(b)	Consider the following concurrent schedule S= $R_2(X)$, $W_3(X)$,$W_1(Y)$,$R_2(Y)$,$W_2(Z)$,$R_4(X)$,$R_4(Y)$ **A)** Check whether the above schedule is conflict serializable or not with required steps using precedence graph. **B)** If the above schedule is conflict serializable, then find all possible equivalent serial schedules .	3+1
	Ans	**Evaluation Scheme:** Full mark for the correct answer. Stepwise mark may be awarded based on the partial correctness of the solution. 2 marks for part A and rest 2 marks for part B. **Solution:** A) Step 1: Create node for each transaction. Step 2: Find the conflict pairs (RW, WR, WW) on same variable by different transactions in reference to below table. <table><tr><th>T1</th><th>T2</th><th>T3</th><th>T4</th></tr><tr><td></td><td>Read(X)</td><td></td><td></td></tr><tr><td></td><td></td><td>Write(X)</td><td></td></tr><tr><td>Write(Y)</td><td></td><td></td><td></td></tr><tr><td></td><td>Read(Y)</td><td></td><td></td></tr><tr><td></td><td>Write(Z)</td><td></td><td></td></tr><tr><td></td><td></td><td></td><td>Read(X)</td></tr></table>	

				Read(Y)

Step 3: Draw edges



Step 4: The precedence graph is acyclic so conflict serializable schedule.

B) The above schedule is a conflict serializable. All the possible topological orderings of the above precedence graph will be the possible serialized schedules. The topological orderings can be found by performing the topological sort of the above precedence graph. After performing the topological sort, the possible serialized schedules are:

T1→T2→T3→T4

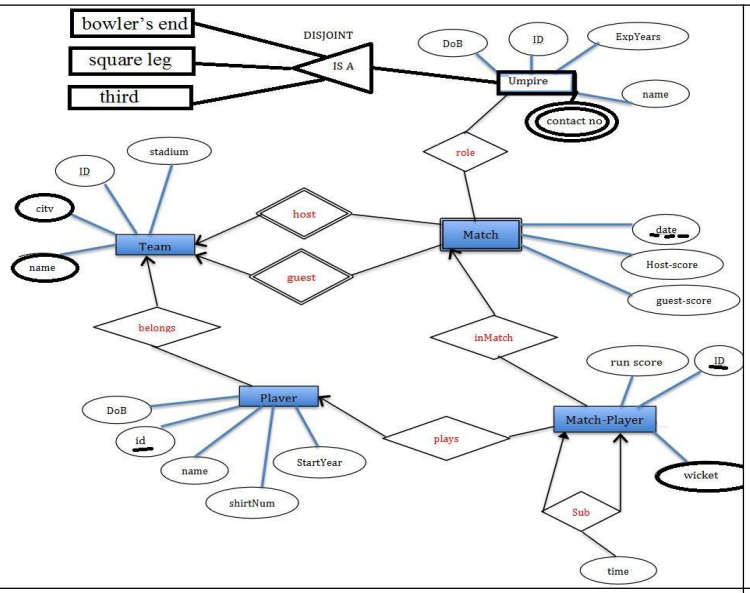
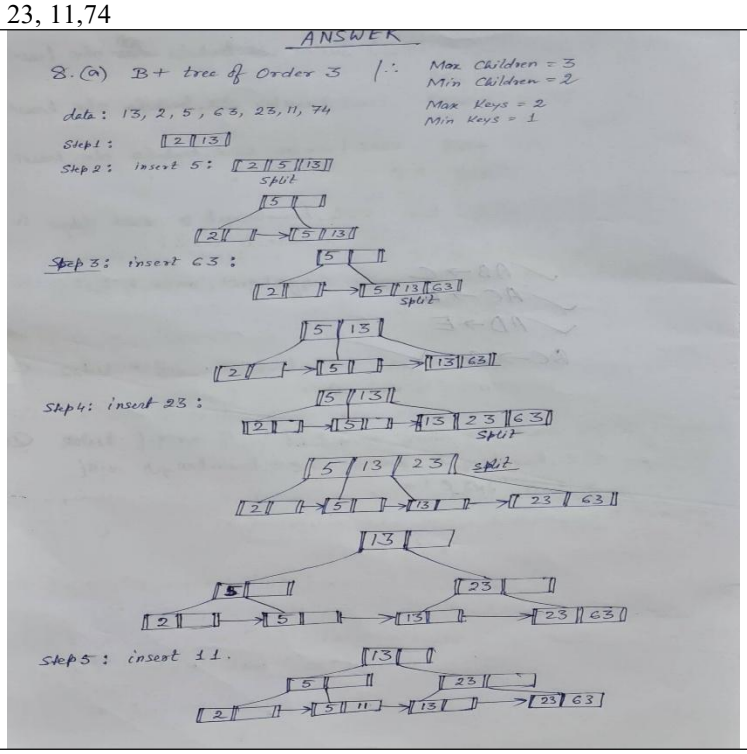
SECTION-D (Learning levels 4,5,6)

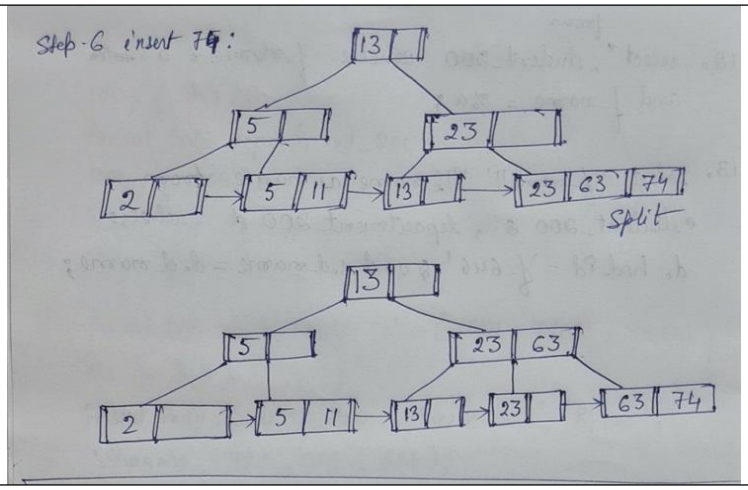
7. (a) Assume there are different IPL cricket teams, having players in each team. In the ER design, we want to show the following: [4]

There are a set of teams, each team has an ID (unique identifier), name stadium_name, and to which city this team belongs.
Each team has many players, and each player belongs to one team. Each player has a number (unique identifier), player_name, DoB, age(as derived attribute) start year, and shirt number that he uses.
Teams play matches, in each match there is a host team and a guest team. The match takes place in the stadium of the host team.
For each match we need to keep track of the following:

- The date on which the game is played
- The final result of the match
- The players participated in the match.
- For each player, how many runs he scored, whether or not he took any wicket
- During the match, one player may substitute another player. We want to capture this substitution and the time at which it took place.
- Each match has exactly three umpires (umpire can either be a bowler's end umpire, square leg umpire, third umpire). For each umpair we have an ID (unique identifier), name, contact number (as multi-valued attribute) DoB, years of experience.

Design an ER diagram to capture the above requirements. State any assumptions you have that affect your design. Clearly specify the cardinalities and primary keys.

	Ans		
	(b)	Map the ERD in above question to create the relational model. (Indicate primary keys as well as referential integrity constraint).	[4]
	Ans	Team=(ID, name, city, stadium_name) Team=(ID, name, city, stadium_name) Match=(ID,date, final result, player participated) Player=(number, dob, player name, age, shirtnum, startyear) Role=(ID,date) Bowler end umpire=(ID, name, year of experience, dob) Contact no=(ID,cont_no) Square leg umpire=(ID, name, year of experience, dob) Third umpire=(ID, name, year of experience, dob) Match player= (ID, run score, wicket time)	
8.	(a)	Create a B+ tree of order 3, with the following data: 13, 2, 5, 63, 23, 11,74	[2]
	Ans		



(b) A relation R(A,B,C,D,E,G) and FD: AB->C, AC-> B, AD->E, B->D, BC->A, E->G and decompose R into R1(A,B,C) R2(A,D,C,E) R3(A,D,G) [6]

- Find out the candidate keys of relation R.
- The above decomposition is lossy or lossless? Justify your answer.
- The above decomposition is dependency preserving or not? Justify your answer.

Ans (b) Candidate keys: AB, AC, BC

	A (A1)	B (A2)	C (A3)	D (A4)	E (A5)	G A(6)
R1	a1	a2	a3	b14	b15	b16
R2	a1	b22	a3	a4	a5	b26
R3	a1	b32	b33	a4	b35	a6

For AB→C, no change

For AC→B,

	A (A1)	B (A2)	C (A3)	D (A4)	E (A5)	G A(6)
R1	**a1**	**a2**	**a3**	b14	b15	b16
R2	**a1**	~~**b22**~~ **a2**	**a3**	a4	a5	b26
R3	a1	b32	b33	a4	b35	a6

For AD→E,

	A (A1)	B (A2)	C (A3)	D (A4)	E (A5)	G A(6)
R1	a1	a2	a3	b14	b15	b16
R2	**a1**	a2	a3	**a4**	a5	b26
R3	**a1**	b32	b33	**a4**	~~**b35**~~ **a5**	a6

For B→D,

	A (A1)	B (A2)	C (A3)	D (A4)	E (A5)	G A(6)
R1	a1	**a2**	a3	~~**b14**~~ **a4**	b15	b16
R2	a1	**a2**	a3	**a4**	a5	b26
R3	a1	b32	b33	a4	a5	a6

For BC→A, no change

For E→G,

	A (A1)	B (A2)	C (A3)	D (A4)	E (A5)	G A(6)
R1	a1	a2	a3	a4	b15	b16

			R2	a1	a2	a3	a4	**a5**	**~~b26~~ a6**	
			R3	a1	b32	b33	a4	**a5**	**a6**	
		The decomposition is lossless. It is not dependency preserving. FDs, B→D and E→G are lost after the decomposition.								
		*****								