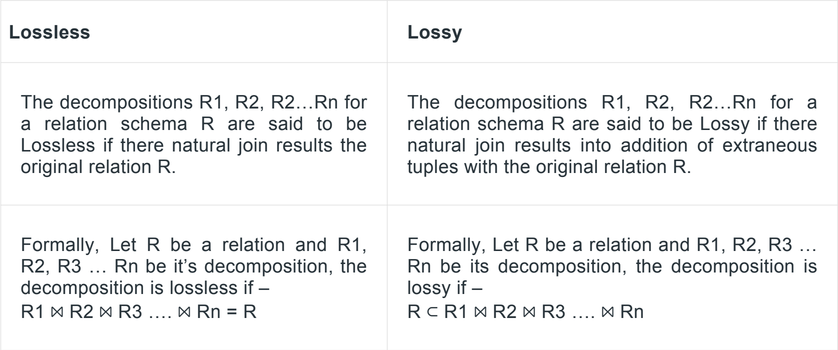
R V College of Engineering
Department of Computer Science and Engineering
CIE - III(Improvement): Scheme

Subject : (Code)	Database Management Systems (CD252IA)	Semester : 5TH BE

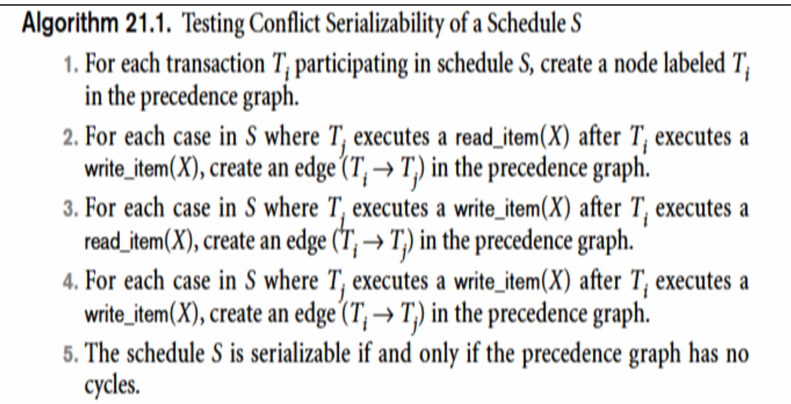
<table>
<tr><th>S.N</th><th>PART-A</th><th>M</th><th>BT</th><th>Co</th></tr>
<tr><td>1.</td><td>What is the difference between lossless and lossy decomposition in DBMS?



<table>
<tr><th>Lossless</th><th>Lossy</th></tr>
<tr><td>The decompositions R1, R2, R2…Rn for a relation schema R are said to be Lossless if there natural join results the original relation R.</td><td>The decompositions R1, R2, R2…Rn for a relation schema R are said to be Lossy if there natural join results into addition of extraneous tuples with the original relation R.</td></tr>
<tr><td>Formally, Let R be a relation and R1, R2, R3 … Rn be it's decomposition, the decomposition is lossless if –
R1 ⋈ R2 ⋈ R3 …. ⋈ Rn = R</td><td>Formally, Let R be a relation and R1, R2, R3 … Rn be its decomposition, the decomposition is lossy if –
R ⊂ R1 ⋈ R2 ⋈ R3 …. ⋈ Rn</td></tr>
</table>
</td><td>2</td><td>L2</td><td>3</td></tr>
<tr><td>2.</td><td>List the two conditions for checking the Binary decomposition?
▪ The FD ((R1 ∩ R2) → (R1 – R2)) is in F+,
▪ The FD ((R1 ∩ R2) → (R2 – R1)) is in F+</td><td>2</td><td>L1</td><td>2</td></tr>
<tr><td>3.</td><td>Define the Condition of 3NF?
Third normal form (3NF) is based on the concept of transitive dependency. A functional dependency X → Y in a relation schema R is a transitive dependency if there exists a set of attributes Z in R that is neither a candidate key nor a subset of any key of R, and both X → Z and Z → Y hold.</td><td>2</td><td>L1</td><td>2</td></tr>
<tr><td>4.</td><td>Define a Transaction with example.
A transaction is an executing program that forms a logical unit of database processing. A transaction includes one or more database access operations—these caninclude insertion, deletion, modification, or retrieval operations.
Ex: airline reservation systems</td><td>2</td><td>L1</td><td>1</td></tr>
<tr><td>5.</td><td>Elaborate and Define ACID properties:
Atomicity, Consistency,Isolation, Durability----0.5m each</td><td>2</td><td>L1</td><td>1</td></tr>
<tr><td></td><td><b>PART-B</b></td><td></td><td></td><td></td></tr>
<tr><td>1a</td><td>Discuss the condition for two functional dependencies to be equivalent? Check whether relation R(A,B,C,D) having two FD sets FD1 = {A->B, B->C, AB->D} and FD2 = {A->B, B->C, A->C, A->D} are equivalent or not ?

Condition: Two sets of functional dependencies E and F are equivalent if E+ = F+. Therefore, equivalence means that every FD in E can be inferred from F, and every FD in F can be inferred from E; that is, E is equivalent to F if both the conditions—E covers F and F covers E—hold---1m
Step 1: Checking whether all FDs of FD1 are present in FD2
• A->B in set FD1 is present in set FD2.
• B->C in set FD1 is also present in set FD2.</td><td>5</td><td>L3</td><td>2</td></tr>
</table>

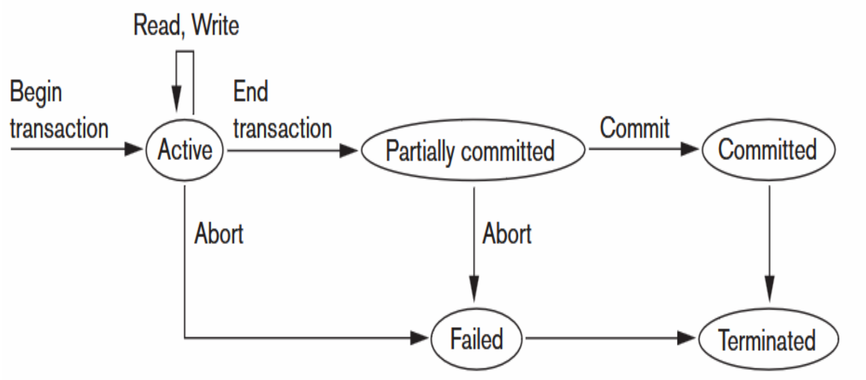
	• AB->D is present in set FD1 but not directly in FD2 but we will check whether we can derive it or not. For set FD2, (AB)+ = {A, B, C, D}. It means that AB can functionally determine A, B, C, and D. So AB->D will also hold in set FD2. As all FDs in set FD1 also hold in set FD2, FD2 FD1 is true. Step 2: Checking whether all FDs of FD2 are present in FD1 • A->B in set FD2 is present in set FD1. • B->C in set FD2 is also present in set FD1. • A->C is present in FD2 but not directly in FD1 but we will check whether we can derive it or not. For set FD1, (A)+ = {A, B, C, D}. It means that A can functionally determine A, B, C, and D. SO A->C will also hold in set FD1. • A->D is present in FD2 but not directly in FD1 but we will check whether we can derive it or not. For set FD1, (A)+ = {A, B, C, D}. It means that A can functionally determine A, B, C, and D. SO A->D will also hold in set FD1. As all FDs in set FD2 also hold in set FD1, FD1 FD2 is true. As FD2 FD1 and FD1 FD2 both are true **FD2 =FD1** is true. These two FD sets are semantically equivalent. -----4m			
1b	Explain any 5 reasons for failure of transaction. A transaction or system error Local errors or exception conditions detected by the transaction, A computer failure (system crash). Concurrency control enforcement. Disk failure. Physical problems and catastrophes.-----any 5 from above -----1m each	5	L2	
2a	Explain the steps for finding Minimal Cover for Functional Dependencies. For the given set of FDs {A->C, AC->D, E->H, E->AD} find the minimal cover. ● Steps: we define a set of functional dependencies F to be minimal if it satisfies the following conditions: ● 1. Split the right-hand attributes of all FDs:Every dependency in F has a single attribute for its right-hand side. ● 2. Remove all redundant FDs. ● 3. Find the Extraneous attribute and remove it.. ● Example: Minimize {A->C, AC->D, E->H, E->AD} ● **Step 1**: {A->C, AC->D, E->H, E->A, E->D} ● **Step2**:{A->C,AC->D,E->H,E->A} Here Redundant FD : {E->D} ● **Step3**:{AC->D}{A}+={A,C} Therefore C is extraneous and is removed. {A->D} ● Minimal Cover = {A->C, A->D, E->H, E->A}	7	L3	2
2b	Write the algorithm for Testing whether a schedule is serializable or not.	3		

<table>
<tr><td></td><td>

Algorithm 21.1. Testing Conflict Serializability of a Schedule S

1. For each transaction T_i participating in schedule S, create a node labeled T_i in the precedence graph.
2. For each case in S where T_j executes a read_item(X) after T_i executes a write_item(X), create an edge ($T_i \rightarrow T_j$) in the precedence graph.
3. For each case in S where T_j executes a write_item(X) after T_i executes a read_item(X), create an edge ($T_i \rightarrow T_j$) in the precedence graph.
4. For each case in S where T_j executes a write_item(X) after T_i executes a write_item(X), create an edge ($T_i \rightarrow T_j$) in the precedence graph.
5. The schedule S is serializable if and only if the precedence graph has no cycles.</td><td></td><td></td><td></td></tr>
<tr><td>3a</td><td>Explain the properties of Attribute preservation and dependency preservation?
Attribute preservation---2.5m
Dependency preservation-2.5m</td><td>5</td><td>L2</td><td>3</td></tr>
<tr><td>3b</td><td>Given a relational schema R = { SSN, ENAME, PNUMBER, PNAME, PLOCATION, HOURS } and the decomposed table R1 = { ENAME, PLOCATION } and R2 = { SSN, PNUMBER, HOURS, PNAME, PLOCATION } and FD = { SSN → ENAME, PNUMBER → { PNAME, PLOCATION}, { SSN, PNUMBER } → HOURS }. Identify whether the given decomposition of R, R1 and R2 is lossless or lossy decomposition ?
Matrix:
<table>
<tr><td></td><td>SSN</td><td>ENAME</td><td>PNUMBER</td><td>PNAME</td><td>PLOCATION</td><td>HOURS</td></tr>
<tr><td>R1</td><td></td><td>a</td><td></td><td></td><td>a</td><td></td></tr>
<tr><td>R2</td><td>a</td><td></td><td>a</td><td>a</td><td>a</td><td>a</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
Final matrix
<table>
<tr><td></td><td>SSN</td><td>ENAME</td><td>PNUMBER</td><td>PNAME</td><td>PLOCATION</td><td>HOURS</td></tr>
<tr><td>R1</td><td>b</td><td>a</td><td>b</td><td>b</td><td>a</td><td>b</td></tr>
<tr><td>R2</td><td>a</td><td>b</td><td>a</td><td>a</td><td>a</td><td>a</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
\
It’s a lossy decomposition</td><td>5</td><td>L3</td><td></td></tr>
<tr><td>4a</td><td>Given a relation R(A, B, C, D) and Functional Dependency set FD = { AB → CD, B → C }, determine whether the given R is in 2NF? If not convert it into 2 NF.
Solution: No non-prime attribute should be partially dependent on Candidate Key Since R has 4 attributes: - A, B, C, D, and Candidate Key is AB, Therefore, prime attributes (part of candidate key) are A and B while a non-prime attribute are C and D
a) FD: **AB → CD** satisfies the definition of 2NF, that non-prime attribute(C and D) are fully dependent on candidate key AB
b) FD: **B → C** does not satisfy the definition of 2NF, as a non-prime attribute(C) is partially dependent on candidate key AB(i.e. key should not be broken at any cost)
As FD B → C, the above table R(A, B, C, D) is not in 2NF---3M

Conversion to 2NF:
a) R1(B, C)
b) R2(A, B, D)-----2M</td><td>5</td><td>L3</td><td>1</td></tr>
<tr><td>b</td><td>With a transition diagram explain the states for transaction execution</td><td>5</td><td>L2</td><td>2</td></tr>
</table>

	 Read, Write Begin transaction Active End transaction Partially committed Commit Committed Abort Abort Failed Terminated Diag-2m Expl-3m			
5.	List and explain with examples the types of problems that can be encountered if two transactions are executing concurrently. ● The Lost Update Problem. ● The Temporary Update (or Dirty Read) Problem. ● The Incorrect Summary Problem. ● The Unrepeatable Read Problem. List -2m Explanation- 8m	10	L2	1